

---

# HP NonStop CORBA 2.6.1 Getting Started Guide for Java

Part number: 525934–004. Published August 2009.

[Legal Notice](#)

## Abstract

This manual provides information about getting HP NonStop™ CORBA® 2.6.1 for Java installed and running. The information provided focuses on the HP-specific implementation of the Object Management Group's CORBA standards.

*Product Version:* HP NonStop™ CORBA® 2.6.1 C11

---

## Table of Contents

[New and Changed Information](#)

[About This Guide](#)

[Who Should Read This Guide](#)

[Organization of this Guide](#)

[Manuals in the NonStop CORBA Set](#)

[Notation Conventions](#)

1. [Introduction to NonStop CORBA](#)

[Why NonStop CORBA](#)

[CORBA 2.6.1 Base](#)

[Client/Server Computing with NonStop CORBA](#)

[Features of NonStop CORBA](#)

[Scalability of the Object Request Broker \(ORB\)](#)

[Scalability of Application Server Processes](#)

[Compatibility of NonStop CORBA Components](#)

[Availability and Fault Tolerance](#)

[Transaction Protection and Data Integrity](#)

[Network Session Concentration](#)

[Advantages of the TS/MP Environment](#)

[IIOB and GIOP Protocol Support](#)

[IIOB/SSL Protocol Support](#)

[IDL Compiler](#)

[Language Bindings](#)

[Portable Object Adapter](#)

[Object Method Invocation](#)

[Naming Service](#)

[Bootstrap Daemon \(BSD\)](#)

[Java Transaction Service \(JTS\)](#)

[NSots XID Broker](#)

[Object by Value](#)

[Event Service](#)

[Error Logging and Tracing](#)

[Portable Interceptors](#)

[Legacy Wrapper Interface](#)

## 2. [Installing NonStop CORBA](#)

[Installation Overview](#)

[Before You Begin the Installation](#)

[Overview of Installation Steps](#)

[Hardware and Software Requirements](#)

[Check for Previously Installed Products](#)

[HP NonStop Server Toolkit for BEA WebLogic Server](#)

[NonStop Enterprise Java Beans \(EJB\)](#)

[Perform Subsystem Resource Cleanup](#)

[Install NonStop CORBA on the Host System](#)

[Run the IPSetup Program](#)

[Run the DSM/SCM Planner Interface](#)

[Change Ownership of OSS Files \(if Using DSM/SCM\)](#)

[Run the NonStop CORBA OSS Installer Setup Program](#)

[Host Installation Installer Wizard](#)

[Install the NonStop Distributed Component Console on the PC](#)

[Install JORB Lite on the PC](#)

[Start NonStop CORBA Server Processes on the Host System](#)

[Modifying an Existing NonStop CORBA 2.6.1 Installation to Add the IIOP/SSL features](#)

[Installing IIOP/SSL on an Existing NonStop CORBA 2.6.1 System](#)

[Alternative Installation Method](#)

[Run the Install Script](#)

[Sample Script Run](#)

[Installing Multiple Copies of NonStop CORBA 2.6.1 on the Same Server](#)

### 3. [Considerations for Migrating from Previous Releases](#)

[Product Level Migration Information](#)

[Overview](#)

[Upgrading From NonStop CORBA 2.3 to NonStop CORBA 2.6.1](#)

[Moving Applications from NonStop CORBA 2.3 Update 3 or Later to NonStop CORBA 2.6.1](#)

### 4. [Running the Java Stack Example](#)

[The Stack Example Overview](#)

[Building the Stack Sample](#)

[Configuring NonStop CORBA for the Stack Example](#)

[Running the Stack Example on OSS](#)

### 5. [Installing JORB Lite](#)

[JORB Lite Overview](#)

[The JORB Lite Package](#)

[Installing JORB Lite](#)

[The JORB Lite Directory Structure](#)

[The PATH and CLASSPATH Environment Variables](#)

[Running the IDL Compiler](#)

### 6. [Running JORB Lite](#)

[Building and Running the Stack Example](#)

[Building the Stack Example](#)

[Running the Stack Example](#)

[Calling NonStop CORBA Applications from JORB Lite](#)

[Running Clients and Servers on Remote Machines](#)

[Index](#)



© Copyright 2009 Hewlett-Packard Development Company L.P.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Export of this documentation may require authorization from the U.S. Department of Commerce.

Microsoft, Windows, and Windows NT are U.S. registered trademarks of Microsoft Corporation.

Intel, Itanium, Pentium, and Celeron are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java is a U.S. trademark of Sun Microsystems, Inc.

Motif, OSF/1, UNIX, X/Open, and the "X" device are registered trademarks, and IT DialTone and The Open Group are trademarks of The Open Group in the U.S. and other countries.

Open Software Foundation, OSF, the OSF logo, OSF/1, OSF/Motif, and Motif are trademarks of the Open Software Foundation, Inc.

OSF MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THE OSF MATERIAL PROVIDED HEREIN, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

OSF shall not be liable for errors contained herein or for incidental consequential damages in connection with the furnishing, performance, or use of this material.

© 1990, 1991, 1992, 1993 Open Software Foundation, Inc. This documentation and the software to which it relates are derived in part from materials supplied by the following:

© 1987, 1988, 1989 Carnegie-Mellon University. © 1989, 1990, 1991 Digital Equipment Corporation. © 1985, 1988, 1989, 1990 Encore Computer Corporation. © 1988 Free Software Foundation, Inc. © 1987, 1988, 1989, 1990, 1991 Hewlett-Packard Company. © 1985, 1987, 1988, 1989, 1990, 1991, 1992 International Business Machines Corporation. © 1988, 1989 Massachusetts Institute of Technology. © 1988, 1989, 1990 Mentat Inc. © 1988 Microsoft Corporation. © 1987, 1988, 1989, 1990, 1991, 1992 SecureWare, Inc. © 1990, 1991 Siemens Nixdorf Informationssysteme AG. © 1986, 1989, 1996, 1997 Sun Microsystems, Inc. © 1989, 1990, 1991 Transarc Corporation.

This software and documentation are based in part on the Fourth Berkeley Software

Distribution under license from The Regents of the University of California. OSF acknowledges the following individuals and institutions for their role in its development: Kenneth C.R.C. Arnold, Gregory S. Couch, Conrad C. Huang, Ed James, Symmetric Computer Systems, Robert Elz. © 1980, 1981, 1982, 1983, 1985, 1986, 1987, 1988, 1989 Regents of the University of California.

---

# New and Changed Information

The updates in this edition (525934–004) are:

- Updated the [Event Service](#) sub-section in the Introduction chapter.

The updates in previous edition (525934–002) were:

Added the following new features for HP NonStop CORBA 2.6.1:

- Support for IIOP/SSL for JORB.
- Support for NonStop Server for Java 4.0.
- Support for Pthreads (T1248) used by NSJ 4.0.
- Support for HP Enterprise Toolkit (ETK) 2.0.
- Explanation on how to send the IP address to the client.

Installation prerequisites, procedures, and migration information have been updated. For these changes, see Chapters 2 and 3.

# About This Guide

[Who Should Read This Guide](#)

[Organization of this Guide](#)

[Manuals in the NonStop CORBA Set](#)

[Notation Conventions](#)

The *NonStop CORBA Getting Started Guide for Java* covers the material you need to know to get NonStop CORBA up and running. This guide starts with a brief introduction to distributed-object computing and how NonStop CORBA fits into the picture, then gives instructions for installation, steps for starting a configuration, and instructions on how to run a sample program to verify installation. The guide also describes how to install and run JORB Lite.

## Who Should Read This Guide

This guide is intended for CORBA system planners or system administrators who are responsible for installing NonStop CORBA and verifying installation.

- i Note:** This guide describes only the NonStop server differences and additions that allow standard CORBA applications to work on the NonStop platform. The reader must be knowledgeable about Object Management Group's (OMG) CORBA standards and fully familiar with the OMG CORBA specifications, which are located at the following URL:  
[http://www.omg.org/technology/documents/spec\\_catalog.htm](http://www.omg.org/technology/documents/spec_catalog.htm).

## Organization of this Guide

Chapter 1, [Introduction to NonStop CORBA](#), introduces NonStop CORBA and the features it offers.

Chapter 2, [Installing NonStop CORBA](#), describes how to install the NonStop CORBA product.

Chapter 3, [Considerations for Migrating from Previous Releases](#), describes the steps needed to migrate to NonStop CORBA 2.6.1.

Chapter 4, [Running the Java Stack Example](#), describes how you can test the operation of your system by running a simple CORBA program.

Chapter 5, [Installing JORB Lite](#), describes how to install JORB Lite.

Chapter 6, [Running JORB Lite](#), describes how you can test your JORB Lite system setup by running one

of the examples supplied with the JORB Lite package.

## Manuals in the NonStop CORBA Set

The *NonStop CORBA 2.6.1 Getting Started Guide for C++* describes how to install the product and verify installation. It also includes an introduction to the product.

The *NonStop CORBA 2.6.1 Getting Started Guide for Java* describes how to install the product and verify installation. It also includes an introduction to the product.

The *NonStop CORBA 2.6.1 Administration Guide* gives basic configuration information and describes how to use the NonStop Distributed Component Console and the command-line interface to perform configuration tasks.

The *NonStop CORBA 2.6.1 Programmer's Guide for C++* provides information for C++ application programmers about the HP-specific implementation of CORBA.

*NonStop CORBA 2.6.1 Programmer's Guide for Java* provides information for Java application programmers with special emphasis on the HP-specific implementation of CORBA.

The *NonStop CORBA 2.6.1 Programmer's Reference* provides information for both C++ and Java application programmers about the IDL compiler, other utilities, minor codes, and system errors. For C++ programmers, it serves as the reference complement to the *NonStop CORBA 2.6.1 Programmer's Guide for C++*. For Java programmers, it provides IDL to Java mapping. (Reference information for Java interfaces and classes is provided in Javadoc format as part of the product.)

The *NonStop CORBA 2.6.1 Glossary* provides definitions of CORBA terminology, with special emphasis on the NonStop CORBA implementation.

In addition, Javadoc reference material for Java programmers is provided on the product CD, and the HP NonStop Distributed Component Console includes online help.

## Notation Conventions

Syntax Item	Description
<b>Bold text</b>	Bold text in a paragraph indicates a technical term that is defined within the text and also, in some cases, in the <i>NonStop CORBA 2.6.1 Glossary</i> .
UPPERCASE LETTERS	Uppercase letters indicate keywords and reserved words; enter these items exactly as shown. Items not enclosed in brackets are required. For example: MAXATTACH
<i>Italic</i> letters	Items in <i>Italic</i> letters indicate variable items that you supply. Items not enclosed in brackets are required. In the following example, <i>UserDir</i> is a variable directory name, while <code>/projectX</code> must be entered exactly as shown: <i>UserDir</i> /projectX

Brackets enclose optional syntax items. For example:

```
TERM [ \system-name. ]$terminal-name  
INT[ERRUPTS]
```

[ ]

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list may be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
LIGHTS [ ON  
        [ OFF  
        [ SMOOTH [ num ] ]  
K [ X | D ] address-1
```

{ }

A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list may be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS { $appl-mgr-name }  
                  { $process-name }  
ALLOWSU { ON | OFF }
```

|

A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

...

```
M address-1 [ , new-value ]...  
[ - ] { 0|1|2|3|4|5|6|7|8|9 }...
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char..."
```

-

A dash in a command usually indicates a separate flag.

" "

When quotation marks surround a defined syntax symbol (such as a bracket or brace), they symbolize an actual character that must be entered as shown. For example:

```
"[ repetition-constant-list ]"
```

Other punctuation

Punctuation not previously described (such as parentheses, commas, and semicolons) must be entered as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;  
LISTOPENS SU $process-name.#su-name
```

## Long Commands

If the syntax of a command is too long to be shown on a single line, each continuation line is indented three spaces, the first of which is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] CONTROLLER  
    [ , attribute-spec ]...
```

---

[« previous](#)[up ^](#)[next »](#)[New and Changed Information](#)[home](#)[Chapter 1. Introduction to NonStop  
CORBA](#)

# Chapter 1. Introduction to NonStop CORBA

## [Why NonStop CORBA](#)

[CORBA 2.6.1 Base](#)

[Client/Server Computing with NonStop CORBA](#)

## [Features of NonStop CORBA](#)

[Scalability of the Object Request Broker \(ORB\)](#)

[Scalability of Application Server Processes](#)

[Compatibility of NonStop CORBA Components](#)

[Availability and Fault Tolerance](#)

[Transaction Protection and Data Integrity](#)

[Network Session Concentration](#)

[Advantages of the TS/MP Environment](#)

[IIOP and GIOP Protocol Support](#)

[IIOP/SSL Protocol Support](#)

[IDL Compiler](#)

[Language Bindings](#)

[Portable Object Adapter](#)

[Object Method Invocation](#)

[Naming Service](#)

[Bootstrap Daemon \(BSD\)](#)

[Java Transaction Service \(JTS\)](#)

[NSots XID Broker](#)

[Object by Value](#)

[Event Service](#)

[Error Logging and Tracing](#)

[Portable Interceptors](#)

[Legacy Wrapper Interface](#)

NonStop CORBA provides an infrastructure and development environment that enables you to develop distributed object applications and components that run on the HP NonStop Kernel operating system. The NonStop CORBA infrastructure provides the services and tools to help software developers build object-oriented components and distributed object systems using either the C++ or the Java programming language. These systems can be implemented at the application level, the system level, or as middleware software.

NonStop CORBA 2.6.1 is an implementation of the Common Object Request Broker Architecture (CORBA) defined and promoted by the Object Management Group (OMG). Because NonStop CORBA 2.6.1 is based on CORBA 2.6.1 as defined by the OMG, application clients and components you develop using NonStop CORBA can interoperate with other CORBA servers running on different platforms.

This topic discusses the advantages and practical challenges of distributed object computing and shows how NonStop CORBA and its underlying technologies meet these challenges.

## Why NonStop CORBA

NonStop CORBA is a robust implementation of OMG's Object Management Architecture (OMA) that can be used in large-scale, enterprise-wide, mission-critical systems.

The NonStop CORBA system architecture combines the flexibility of object technology with the robustness of a transaction-processing monitor. This unique combination provides the availability and scalability required for mission-critical applications. In addition, NonStop CORBA ensures the integrity of its own data stores and offers an object transaction service you can use to maintain a secure environment for your applications.

NonStop CORBA gives you a CORBA 2.6.1-based, object-oriented development system that provides differentiation in the areas of scalability, availability, and data integrity (transaction protection). Due to tight integration between HP's transaction services and transaction monitors, NonStop CORBA contains the true functionality of an object transaction monitor.

Scalability allows a system to grow dynamically as usage increases. NonStop CORBA provides scalability in the following dimensions:

- Network connections
- Object Request Broker (ORB) processes
- Application processes

Optimized for use in transaction processing, NonStop CORBA allows the mapping of a large number of clients to a smaller number of servers, effectively sharing resources. Because the NonStop CORBA system processes run in an HP NonStop TS/MP environment, you get the strength of HP transaction services (for process management, availability, and load balancing) in a CORBA-compliant-based environment.

## CORBA 2.6.1 Base

New features of NonStop CORBA are based on the CORBA 2.6.1 specification of the Object Management Group (OMG) and other OMG specifications, except as noted in the *NonStop CORBA 2.6.1 Programmer's Reference*. Being CORBA-based means that NonStop CORBA interoperates with other CORBA Object Request Brokers (ORBs). By using the Internet Inter-ORB Protocol (IIOP™) defined by CORBA, NonStop CORBA can interoperate with other heterogeneous OMA-based environments that use the CORBA IIOP protocol.

Writing client and server objects compliant with CORBA 2.6 offers the following benefits:

- Interoperation with other CORBA-compliant ORBs
- Wide area network (WAN) and local area (LAN) network connectivity based on international networking standards
- Ability to wrap legacy applications with a distributed object architecture (an advantage of NonStop CORBA)

## Client/Server Computing with NonStop CORBA

NonStop CORBA applications are client/server applications that run on Common Object Request Broker Architecture (CORBA) 2.6-based computer systems and networks. While clients and servers can be running on the same system, NonStop CORBA servers can also handle remote client requests. Also, NonStop CORBA clients can make requests that are fulfilled by remote NonStop CORBA servers and servers on other ORBs™.

NonStop CORBA application servers are CORBA-based server processes that run on the HP NonStop Kernel operating system. The application server acts as a server process for clients both local and remote to the NonStop CORBA system.

NonStop CORBA clients are CORBA client processes that also run on the NonStop Kernel operating system. As CORBA clients, they can either reside on the same system as the NonStop CORBA application server that processes their requests, or they can act as "network" or "remote" clients by interoperating across a network to an application server located on another computer that may use another vendor's CORBA-compliant ORB.

With NonStop CORBA, a network client need not be aware of where the application server is located; when a network client makes a request of an object, the NonStop CORBA software transparently establishes a link with the server to handle the client's request.

## Features of NonStop CORBA

By implementing certain system-specific processes using TS/MP server classes or server pools (the preferred terminology), the ORB provided with NonStop CORBA offers availability and scalability advantages over other CORBA-based ORBs. TS/MP runtime code resides in every ORB running NonStop CORBA and can also be included in your NonStop CORBA application servers.

The NonStop CORBA ORB runs on HP NonStop systems. The NonStop CORBA application servers that you write in C++ or Java use this ORB for communication to and from remote clients. The major features differentiating the ORB supplied with NonStop CORBA from other ORBs are:

- [Scalability of the ORB](#) to support higher throughput
- [Scalability of application server processes](#) across CPUs
- [Compatibility](#) of NonStop CORBA components
- [Availability and fault tolerance](#)
- [Transaction protection](#) for data stores managed by the NonStop TM/MP and database software
- [Network session concentration](#) to reduce the number of IP ports needed in the host
- [Advantages of the TS/MP environment](#)

## Scalability of the Object Request Broker (ORB)

With NonStop CORBA, you can increase the ORB capacity without disturbing running applications. NonStop CORBA provides several ways of scaling the ORB itself:

- Scalability of the ORB to support higher throughput New clients can connect to the NonStop CORBA ORB with no configuration changes required on the client workstation or on the HP server.
- To support increased request traffic, the NonStop CORBA system administrator can increase the number of Comm Servers.
- To provide additional external port connections, the administrator can add multiple instances of HP NonStop TCP/IP to the NonStop CORBA system. Each instance of NonStop TCP/IP increases the number of available TCP/IP ports and, therefore, the number of clients the ORB can support. Multiple instances of NonStop TCP/IP allow Comm Servers to listen on the same IP address or different IP addresses (port numbers must be unique within the scope of the TCP/IP process).
- The administrator can configure NonStop CORBA to use the Parallel Library TCP/IP product. This product allows a single TCP/IP port to be shared by multiple processes, with each connection being established in round-robin fashion. Using Parallel Library TCP/IP increases capacity (there are more processes available to share the load) and improves fault tolerance (a software or processor failure does not render the address unusable).
- Network session concentration to reduce the number of IP ports needed in the host
- Within the HP system, the NonStop CORBA system uses TS/MP for communication between application processes, Comm Servers, and application servers. This design improves throughput while minimizing the IPC resources needed by the ORB.

You can increase the number of remote clients connected to the NonStop CORBA ORB without requiring configuration changes on either the client workstation or the NonStop CORBA system. When a

remote client first makes a connection to the ORB, the NonStop CORBA system transparently assigns an existing port number to the client.

In practice, a single Comm Server can handle hundreds of clients, and a TCP/IP process can handle hundreds of Comm Servers.

## **Scalability of Application Server Processes**

NonStop CORBA allows the scalability of application server processes by supporting the use of TS/MP server pools. A server pool consists of multiple processes (CPU resources) that implement the same application logic. TS/MP provides load balancing, dividing requests among the servers in the pool.

A client program need not be aware of whether an object class is running in a server pool or an individual server process. In fact, the same server program normally can run in either mode; you specify the preferred mode in a configuration file for each server. (HP recommends the use of server pools to provide availability, autorestart, scalability, and load balancing.)

## **Compatibility of NonStop CORBA Components**

NonStop CORBA provides source code portability for applications written to the OMG CORBA specifications. Most of the HP differentiated features, such as scalability, are transparently available; application servers ported to the HP system do not need to change in order to exploit the scalability of NonStop CORBA.

## **Availability and Fault Tolerance**

To support continuous availability, an object-oriented runtime environment must provide fault tolerance on the following three levels:

- The system platform
- The ORB
- The application components

The NonStop CORBA system provides the first level of fault tolerance by running on the NonStop Kernel operating system, which offers such features as NonStop process pairs, mirrored disk controllers, and fault-tolerant communications subsystems.

The ORB supports fault tolerance by using TS/MP as a process manager for ORB processes. TS/MP provides automatic restart when a server process fails.

Automatic restart is also available to application components that use TS/MP server pools.

## **Transaction Protection and Data Integrity**

The Object Transaction Service (OTS) and Java Transaction Service (JTS), provided as a part of NonStop CORBA, can be relied upon to provide transaction protection and data integrity. The NonStop CORBA JTS and OTS use HP NonStop Transaction Management Facility (TMF) as the underlying

mechanism for transaction protection, keeping transaction protection and data integrity in line with HP fundamentals.

The data stores in the Naming Service and the Interface Repository databases, which contain ORB configuration information, are also protected by NonStop TM/MP and database software.

You can use the JTS and OTS to protect your application-specific data when the application itself needs transaction protection or when the client has requested transaction protection.

## Network Session Concentration

The NonStop CORBA system provides network session concentration to reduce the number of connections to the NonStop system. NonStop CORBA provides network session concentration with the use of Comm Servers. Comm Servers are NonStop CORBA system components that manage the linkages between remote clients and local servers. Comm Servers use the following protocols:

- TS/MP protocol for communication with TS/MP server pools.
- Guardian file system protocol for communication with individual processes on HP NonStop systems.
- Internet Inter-ORB Protocol (IIOP), as defined by the OMG specification for communication with remote components over TCP/IP. A remote client uses a single TCP/IP port number to connect to the ORB, regardless of the number of servers the client uses.

Internet Inter-ORB/Secure Sockets Layer (IIOP/SSL) Protocol to provide confidentiality, authentication, and message integrity over IIOP. IIOP/SSL is an optional feature that must be separately installed.

## Advantages of the TS/MP Environment

When you use the NonStop CORBA software you can gain all of the advantages of the TS/MP environment, including ease of development, manageability, data integrity, and fault tolerance.

- [IIOP protocol support](#)
- [IIOP/SSL protocol support](#)
- [IDL Compiler](#)
- [Language bindings](#)
- [Portable Object Adapter \(POA\)](#)
- [Object Method Invocation](#)
- [Naming Service and Interoperable Naming Service](#)
- [Bootstrap Daemon \(BSD\)](#)

- [OTS/JTS](#)
- [NSots XID Broker](#)
- [Object by Value](#)
- [Event Service](#)
- [Error Logging and Tracing](#)
- [Portable Interceptors](#)
- [Legacy Wrapper Interface](#)

## **IIOP and GIOP Protocol Support**

Internet Inter-ORB Protocol (IIOP) is the standard protocol that different ORBs must use to communicate across the Internet. This protocol is standardized by the OMG. Because the objects that you write in NonStop CORBA comply with IIOP they can be used by other ORBs and clients, and NonStop CORBA clients can use the objects located on other ORBs that support IIOP.

NonStop CORBA 2.6.1 supports GIOP versions 1.0, 1.1, and 1.2. If a client from another ORB uses an earlier version of GIOP, NonStop CORBA automatically adopts that version.

## **IIOP/SSL Protocol Support**

IIOP/SSL extends the support of the IIOP protocol to provide security features at the transport layer. IIOP/SSL adds confidentiality, authentication, and message integrity to IIOP communications. Installation of IIOP/SSL is optional.

## **IDL Compiler**

The NonStop CORBA IDL compiler (NSDIDL) is based on the CORBA 2.6.1 specifications defined by the OMG. Any differences between the specification and the NonStop CORBA implementation are noted in the *NonStop CORBA 2.6.1 Programmer's Reference*. The compiler generates stubs and skeletons for use by client objects and server objects, respectively.

## **Language Bindings**

NSDIDL supports OMG standard C++ and Java language bindings, as defined in the OMG binding specifications. Any differences between the specification and the NonStop CORBA implementation are noted in the *NonStop CORBA 2.6.1 Programmer's Reference*.

## **Portable Object Adapter**

The NonStop CORBA Portable Object Adapter (POA) supports the function defined in the CORBA specification. An Object Adapter is the primary way that an object implementation accesses the services

an ORB provides.

The NonStop CORBA POA:

- Allows for development of portable object solutions
- Provides for objects that persist over long periods of time
- Allows a servant to manage multiple objects as one logical object (A servant is a user-defined object that provides an implementation of the methods of an interface.)
- Allows multiple distinct instances of a POA to exist in a server
- Supports transient objects
- Supports application use of static and/or dynamic skeletons
- Provides an extensible method for associating policies with objects
- Supports threaded objects

Application programmers upgrading from the Basic Object Adapter (BOA) to the POA can rethink design issues because they will probably need to alter their application interfaces to make use of the POA. However, basic BOA usage can easily be mapped to basic POA usage.

## Object Method Invocation

NonStop CORBA supports the Static Invocation Interface (SII).

## Naming Service

The Naming Service provides for the binding of human-readable object names to CORBA objects, and the resolving of these named objects in client requests. The NonStop CORBA Naming Service is based on the CORBA Naming Service 1.2 specification. The Naming Service database in NonStop CORBA is implemented as a structured Enscribe file.

NonStop CORBA also implements the interoperable extension of the Naming Service, allowing users to access an object through a human-readable URL name. This format makes handling objects simpler for the user. In addition, the client no longer must import an IOR URL from the server to invoke methods on that server. The Interoperable Naming Service defines four additions to the Naming Service:

- Defines the name syntax and parsing interface
- Standardizes bootstrapping to a common name context
- Requires `id` and `kind` attribute matching
- Adds support for URL (Uniform Resource Locator) names

The Naming Service installs automatically when you install NonStop CORBA.

## Bootstrap Daemon (BSD)

The Bootstrap Daemon (BSD) is a component of the Interoperable Naming Service. The BSD is needed for interoperability with other ORBs using the Bootstrap protocol. The BSD protocol provides operations for resolving an initial object reference and for listing the supported initial reference ids. Interoperable client ORBs send a request to an initialization agent. The agent replies to the client ORB with the requested information. The IDL interface provides two operations, a `get()`, which accepts an `ObjectId` and a `list()`, which accepts no arguments. The IDL interface for the two operations is as follows:

```
module CORBA{
    interface InitialReferences {
        Object get(in ObjectID id);
        ObjectIdList list();
    };
};
```

For more information on the Bootstrap Daemon, see the *NonStop CORBA 2.6.1 Administration Guide*.

## Java Transaction Service (JTS)

The NonStop Java Transaction Service (JTS) is automatically installed with NonStop CORBA. JTS implements the Java Transaction API defined by Sun Microsystems, Inc. This service supports transaction processing between distributed objects.

## NSots XID Broker

When transaction context is inherited from one process to another, an imported transaction branch is created. A transaction branch is represented by a new TMF transaction identifier called the XID. The root XID and any branch XIDs are not recognized by TMF or SQL as representing the same transaction. This means that if a process such as an application starts a transaction and makes multiple invocations on server processes, multiple XIDs can be created resulting in SQL errors or blocking. It is up to the application designer to avoid this problem. If more than one NSots™ is configured in a server pool, multiple requests on the same transaction branch might be routed through more than one NSots™. This situation is beyond the control of the application designer.

To solve this problem, NSOTS and NSJTS include an XID Broker function. The XID Broker brokers requests for imported transactions within the pool of NSots™s by routing secondary import requests back to the NSots™ that originally imported the transaction. In previous versions of NSOTS and NSJTS, the broker function was provided through a shared file. The current version of NSOTS and NSJTS replaces the file implementation with a singleton process called NSotsXID.

## Object by Value

NonStop CORBA supports Object by Value, allowing applications to pass objects and parameters by value rather than by reference. The receiving side of an object passed by value receives a description of the "state" of the object and then instantiates a new object with that state but having a separate identity from that of the sending side. Once the parameter-passing operation is complete, no relationship exists between the two instances.

## Event Service

The Event Service allows for decoupled communication between suppliers and consumers of events through an event channel. Suppliers and consumers register with the event channel. At registration, the supplier/consumer indicates either the push or pull model. For the push model, the supplier initiates the transfer of the event. For the pull model, the consumer requests the event transfer. NonStop CORBA supports only the generic communication model. The typed communication model is not supported. The Event Service also installs automatically when you install NonStop CORBA.

## Error Logging and Tracing

All server pools and system infrastructure in NonStop CORBA provide user-configurable error logging, tracing, and event-channel messaging. Application programmers may instrument objects that take advantage of these debugging features.

NonStop CORBA also supplies an IDL interface to the Error Logging facility, allowing the developer to log application-specific information.

## Portable Interceptors

NonStop CORBA supports the Portable Interceptors interface as defined in the OMG CORBA 2.6.1 specification.

## Legacy Wrapper Interface

Legacy wrapping is never an easy task because the legacy application code itself may require modification. However, HP's supplied set of generic legacy wrappers act as an integration framework for your legacy application. You do not need to modify your legacy application, and your legacy application can interoperate transparently with existing CORBA objects and components.

---

[<< previous](#)[up ^](#)[next >>](#)[About This Guide](#)[home](#)[Chapter 2. Installing NonStop CORBA](#)

# Chapter 2. Installing NonStop CORBA

## [Installation Overview](#)

[Before You Begin the Installation](#)

[Overview of Installation Steps](#)

[Hardware and Software Requirements](#)

## [Check for Previously Installed Products](#)

[HP NonStop Server Toolkit for BEA WebLogic Server](#)

[NonStop Enterprise Java Beans \(EJB\)](#)

## [Perform Subsystem Resource Cleanup](#)

## [Install NonStop CORBA on the Host System](#)

[Run the IPSetup Program](#)

[Run the DSM/SCM Planner Interface](#)

[Change Ownership of OSS Files \(if Using DSM/SCM\)](#)

[Run the NonStop CORBA OSS Installer Setup Program](#)

[Host Installation Installer Wizard](#)

## [Install the NonStop Distributed Component Console on the PC](#)

## [Install JORB Lite on the PC](#)

## [Start NonStop CORBA Server Processes on the Host System](#)

## [Modifying an Existing NonStop CORBA 2.6.1 Installation to Add the IIOP/SSL features](#)

[Installing IIOP/SSL on an Existing NonStop CORBA 2.6.1 System](#)

## [Alternative Installation Method](#)

[Run the Install Script](#)

[Sample Script Run](#)

## [Installing Multiple Copies of NonStop CORBA 2.6.1 on the Same Server](#)

## Installation Overview

To install the NonStop CORBA product you use installers running on your PC to place the NonStop CORBA components on your HP NonStop S-Series server. As an alternative you can install directly on the NonStop system through the OSS environment. Both installation methods are discussed in this section. After you complete installation on the NonStop system, you use the HP NonStop Distributed Component Console installer to install the console on your PC. You may also install additional optional software on your PC. When installation is complete you run the console software to start the NonStop Services for CORBA processes. Sample programs can then be run to verify your installation.

## Before You Begin the Installation

- Review the readme file on the product CD to make sure you have the correct version for all products installed on your NonStop S-Series server.
- Make sure your site meets the minimum hardware and software requirements, as indicated in [Hardware and Software Requirements](#).
- Check whether previous installations of application servers that use CORBA are already installed on the system where you intend to install NonStop CORBA, as indicated in [Check for Previously Installed Products](#).
- If you have not used the IPSetup program before, you should refer to the *IPSetup User's Guide* on the product CD for information about this installation utility. The file is USRGUIDE . PDF on the CD.
- Decide if you want to use the Distributed Systems Management/Software Configuration Manager (DSM/SCM) software to manage the NonStop CORBA installation on your system. DSM/SCM manages file placement in the OSS file directories as well as in Guardian volumes and subvolumes. You still need to use the NonStop CORBA installer or installation script to install CORBA files after they are placed by DSM/SCM. If you have not used DSM/SCM before, see the *DSM/SCM User's Guide*. Refer to [Hardware and Software Requirements](#) for the specific versions required.
- Decide how you want to install EMS templates. There are two ways to install the templates:
  - Method 1: Use the NonStop CORBA OSS Installer or the installation script to install the templates during the NonStop CORBA installation.
  - Method 2: Use the `INSTALL^TEMPLATES TACL` macro to install the templates after you complete the NonStop CORBA installation.
- Decide if you want to use the NonStop CORBA PC installer or if you will install directly in the OSS environment by using the installation script.
- Consider the security implications of the user ID that you use to perform your installation. The CORBA installers use the default TACL settings. Depending upon how the user's defaults are set, other users may not be able to access NonStop CORBA databases, such as the configuration database.
- Decide on a naming convention for the Business Domains you will use. The Security Domain will be the logon that you use for installation.
- If you are installing NonStop CORBA over a previous installation of NonStop CORBA, you may need to perform a subsystem resource cleanup as described in [Perform Subsystem Resource Cleanup](#). You should also refer to the [Chapter 3, Considerations for Migrating From Previous Releases](#).

## Overview of Installation Steps

- Place the NonStop CORBA software onto your NonStop server as indicated in [Run the IPSetup Program](#).
- If you are using DSM/SCM, run it to place your software and unpx files, as described in [Run DSM/SCM](#).
- If you used DSM/SCM, change the ownership of the OSS files as described in [Change Ownership of OSS Files \(if Using DSM/SCM\)](#).
- Install the NonStop CORBA software on the NonStop S-Series server as described in [Run the NonStop CORBA OSS Installer Setup Program](#). As an option, you can install the IIOP/SSL features at the same time. You can also perform these steps using the installation script as described in [Alternative Installation Method](#).
- Install the Distributed Component Console software on the PC as indicated in [Install the Distributed Component](#)

[Console on the PC.](#)

- Install optional software on the PC, as indicated in [Install NonStop JORB Lite on the PC.](#)
- Start the server processes, as indicated in [Start NonStop CORBA Server Processes on the Host System.](#)
- Run the stack example to verify your installation.

## Hardware and Software Requirements

- A NonStop S72000 server or above running:
  - HP NonStop Kernel operating system, RVU G06.20 or later
  - HP NonStop Kernel Open System Services (OSS)
  - HP NonStop Server for Java, release 3.1 or later
  - HP NonStop Transaction Management Facility (TMF)
  - HP NonStop TS/MP
  - HP Tandem LAN Access Method (TLAM)
  - HP NonStop TCP/IP
  - HP NonStop DSM/SCM (required only if you manage your installation with DSM/SCM)
- A PC workstation running Microsoft Windows 2000 or Windows XP
- JDK version 1.3.1 only. Do not use earlier or later versions.

## Check for Previously Installed Products

If other Independent Products are installed on the NonStop S-Series server where you intend to install NonStop CORBA software, you need to ensure that the port numbers are different for all products. NonStop CORBA assigns port numbers for the AdminServer, Comm Server (TCP/IP), the BSD, ILSD, and LSD during installation. You can change these port numbers after installation. In addition, check the considerations in the following sections for specific cautions.

Some installations of NonStop CORBA versions are incompatible if you use DSM/SCM to install the products:

- You cannot use DSM/SCM to install a CORBA B12 or CORBA B13 and a C11 product at the same time. Any attempt to do so will terminate with a message similar to the following:  

```
ERROR: A duplicate file was found. Product T0539 and Product T2818 both place
file ZDOMC in TSV ZSPIDEF. These two files have different fingerprints.
```

## HP NonStop Server Toolkit for BEA WebLogic Server

If NonStop Server Toolkit for BEA WebLogic Server is installed on the NonStop S-Series server where you intend to install NonStop CORBA software, each product must have unique values for the port numbers.

## NonStop Enterprise Java Beans (EJB)

NonStop CORBA 2.6.1 should not be installed on the same system as NonStop Enterprise Java Beans (EJB). You should uninstall EJB before installing NonStop CORBA 2.6.1.

# Perform Subsystem Resource Cleanup

A subsystem cleanup does not have to be done in all cases, but you may want to do a cleanup if NonStop CORBA is already installed. If you have a system you want to install over or a system to clean up, then performing a subsystem resource cleanup is advisable. A NonStop CORBA subsystem creates resources during installation and setup. It is necessary to run the `unconfigure` script to remove these resources.

The script is divided into two phases. The first phase removes files created during installation. The second phase removes TMF Resource Manager entities created by OTS Transaction Managers during subsystem startup.

## Phase 1

To execute this phase of the script, you must have the subsystem's `env.sh` file sourced into your environment. You either must use the same logon id as you used during the original installation of the subsystem, or you must be logged on as a super user. For example, assuming the subsystem's prefix is "M", a successful first phase looks like this:

```
oss>> unconfigure
nsdstop must be run prior to running this script and
NSDCFGDB and NAMINGDB database files cannot be opened.
```

```
This will PERMANENTLY purge the NSDCFGDB and NAMINGDB databases.
Do you want to proceed (y/n):
```

```
y
$DATA01.MMD0406.NSDCFGDB PURGED.
1 FILE PURGED
$DATA01.MMD0404.NAMINGDB PURGED.
1 FILE PURGED
```

## Phase 2

To execute this phase of the script, you must be logged on as a super user. For example, assuming the subsystem's prefix is "M" and five processes were configured in the subsystem's `otstm` serverclass, a successful second phase looks like this:

```
oss>> unconfigure
This will PERMANENTLY purge the NSDCFGDB and NAMINGDB databases.
Do you want to proceed (y/n):
n
```

```
This will PERMANENTLY delete the TMF Resource Managers.
Do you want to proceed (y/n):
```

```
y
gtacl[9]: unable to propagate all environment variables
deleted RM NSOTSRMTX_MNO0
gtacl[9]: unable to propagate all environment variables
deleted RM NSOTSRMTX_MNO1
gtacl[9]: unable to propagate all environment variables
deleted RM NSOTSRMTX_MNO2
gtacl[9]: unable to propagate all environment variables
deleted RM NSOTSRMTX_MNO3
gtacl[9]: unable to propagate all environment variables
deleted RM NSOTSRMTX_MNO4
gtacl[9]: unable to propagate all environment variables
```

# Install NonStop CORBA on the Host System

The host system is a NonStop S-Series server. The installers run on your workstation.

## Run the IPSetup Program

This section discusses only automated software placement using the IPSetup Windows application. See the *IPSetup User's Guide* for information about manual software placement using FTP and SETUPRUN.

Exit all other Windows applications before starting to place NonStop CORBA software on the host.

1. Open the product CD by double clicking on the CD drive.
2. Click the **View Readme file** button. Setup opens the readme file in Notepad. Be sure to review the entire readme file before proceeding.
3. Click the **Run IPSetup** button to launch IPSetup. The program displays a Welcome screen and a License Agreement screen. To continue the installation, click **Next** on each of these screens.
4. On the Installation Options screen click the **NonStop Kernel RISC** radio button. You may optionally check the box for **Use DSM/SCM**. Click **Next**.
5. On the Product Selection screen, highlight the NonStop CORBA products you want to install. The software developers kit (SDK), the runtime environment kit (RTK), and IOP/SSL are separate products. Click the **Add** button, then click **Next**.
6. Log on to the host that you wish to install on by following the instructions on the Host Information screen. You may use either the system name or the system IP address to log on. Click **Next**.
7. On the Host Target Settings screen you may either accept the default locations for Work and Backup subvolumes or browse to locations of your choice. Click **Next** when you are satisfied with the locations.
8. On the Host File Placement screen you may either accept the default disk locations or browse to locations of your choice. Click **Next** when you are satisfied with the locations.
9. On the Installation Manifest screen review the file locations. You may click Back to go back and change them. When you are satisfied with them click **Next**. This step may take a few minutes to complete.
10. On the Installation Complete screen you may choose the check boxes to view the release documentation and to launch the DSM/SCM planner interface. It is recommended that you review the release documentation before proceeding to install NonStop CORBA.
11. Click **Finish** to complete running IPSetup.

## Run the DSM/SCM Planner Interface

This procedure is optional. If you choose to use DSM/SCM, see the *DSM/SCM User's Guide* for a description of the software configuration process. If you launch the DSM/SCM planner interface directly from IPSetup, note that STARTTMF and STARTSCM must have been run on your NonStop system before you log on to the planner interface. You should complete the Operator Instructions supplied by the DSM/SCM planner before proceeding with the installation.

## Change Ownership of OSS Files (if Using DSM/SCM)

If you are using DSM/SCM to manage your installation, you need to change the ownership of OSS files at this point in the installation. If you are not using DSM/SCM, you should skip this step.

DSM/SCM allows you to set the default ownership of installed Guardian files. You cannot, however, control the ownership of OSS files. New OSS files are owned by SUPER. SUPER. Installed files that replace existing files inherit the

ownership of the previously installed files. After you finish running the DSM/SCM planner interface, you must change the ownership of NonStop CORBA files as follows:

1. Log on as SUPER.SUPER and start the OSS environment
2. Type this command:  

```
chown -R user.user1 /usr/tandem/nsdoms
```

where *user.user1* is the user ID that will own the NonStop CORBA OSS files.
3. Check the security of the files in `/usr/tandem/nsdoms` to be sure that it is compatible with your system management and organization policies.

## Run the NonStop CORBA OSS Installer Setup Program


Use the NonStop CORBA OSS Installer Setup Program to install the NonStop CORBA OSS Installer program on your PC. You may choose to install the Run Time Kit (RTK) or the Software Developer's Kit (SDK) at this time. As an option, the IOP/SSL features can be installed as part of the RTK or SDK. When finished, the NonStop CORBA/OSS Installer wizard starts automatically to install NonStop CORBA software on the host system.

1. To start the NonStop CORBA OSS Installer Setup Program on your workstation (`\WS_SW` directory), double-click the `CORBA_INSTALL.EXE` file on your product CD. The Setup Program displays a Welcome screen and a screen that describes the setup program. To continue the installation, click **Next** on each of these screens.
2. On the Choose Destination Location screen, choose a destination folder in which to install the NonStop CORBA OSS Installer or accept the default location of `c:\nsorb`. Click **Next** to continue.
3. On the Select a Program Manager Group screen, select a Program Manager Group to contain the NonStop CORBA OSS Installer icon or accept the default group of NonStop CORBA. Click **Next** to continue.
4. If the installer cannot find the `javaw.exe` file (usually located in the JRE), the Java Runtime Environment (JRE) installation dialog box appears. In this dialog, you can specify where the JRE installation that contains `javaw.exe` is located. Otherwise, you can choose to install the JRE on your workstation.
5. The Start Installation screen indicates you are ready to start the NonStop CORBA OSS Installer installation. Click **Next**.
6. If you have chosen to install the JRE in [Step 4](#), the Java Runtime Environment Setup program will take over and you will be asked to agree to the Java licensing agreement. Click **Yes** to continue. As the installation runs, the installation progress is displayed on the Installing screen. When the JRE installation is complete you will be returned to the NonStop CORBA OSS Installer Setup Program.
7. When the NonStop CORBA OSS Installer Setup Program is finished, the Installation Complete screen appears. Click **Finish**.

## Host Installation Installer Wizard

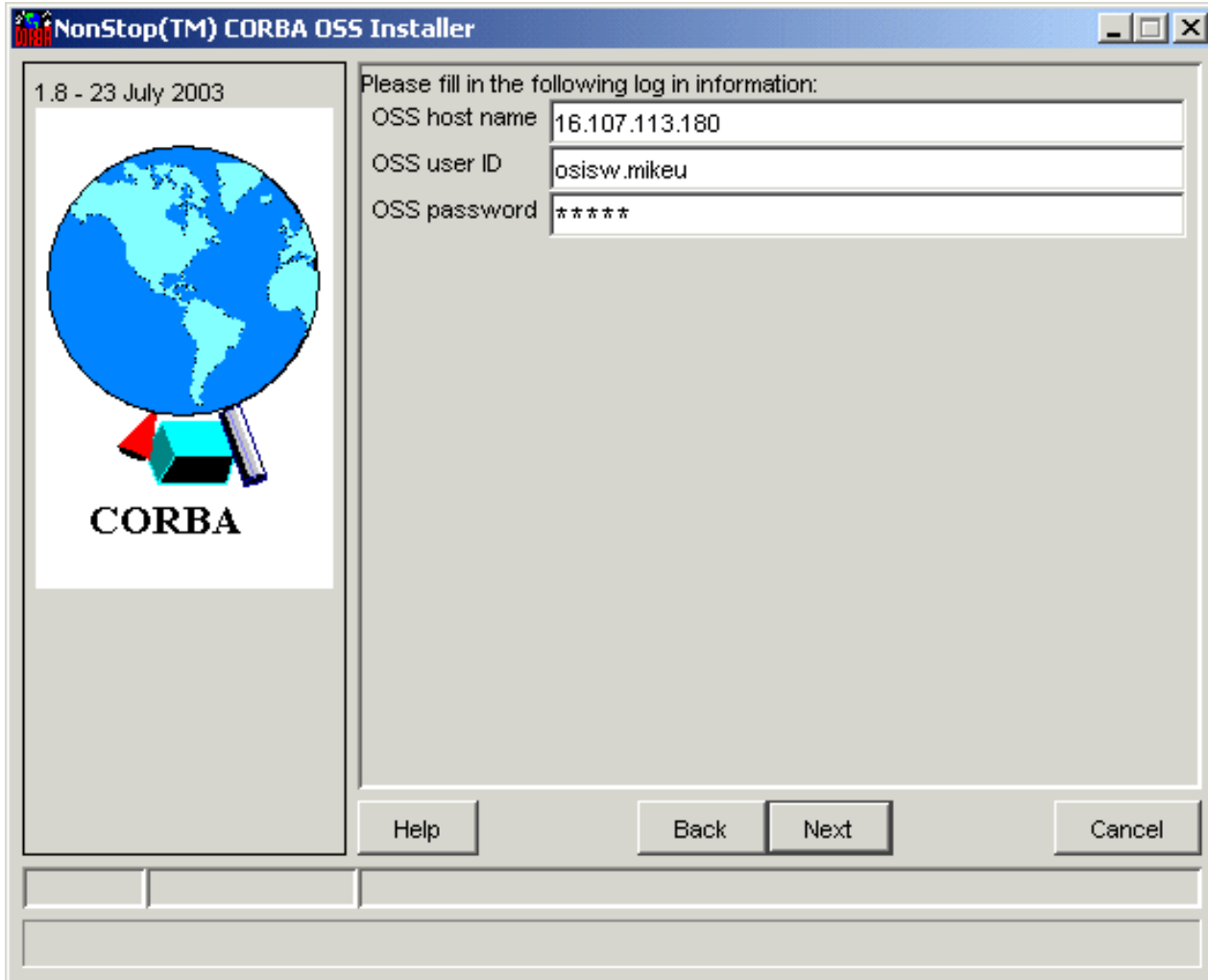
1. The NonStop CORBA OSS Installer wizard starts automatically when you click Finish on the setup program and displays a Welcome screen. To continue the installation, click **Next**.

To install NonStop CORBA later, click **Cancel**. The installer program is at `Start>Programs>NonStop(TM) CORBA>OSS Installer`.

2. After the Welcome screen, a screen appears that summarizes what the installer will do. Click **Next**.  
 **Note:** To do this installation, you must have already installed NonStop Server for Java and have write access to its directories.
3. Type values for the OSS `host name`, OSS `user ID`, and OSS `password` fields. Click **Next** to continue.

**i Note:** If you used the DSM/SCM planner interface to move your files onto the NonStop system and unpax them, the host name, user ID, and password are provided as part of the Operator Instructions in DSM/SCM.

Field Name	Description
OSS host name	Name or IP address of the host system.
OSS user ID	User ID having write permission on the OSS directory for pax file (Guardian space) and Java OSS root directory.
OSS password	Password for the OSS user ID.



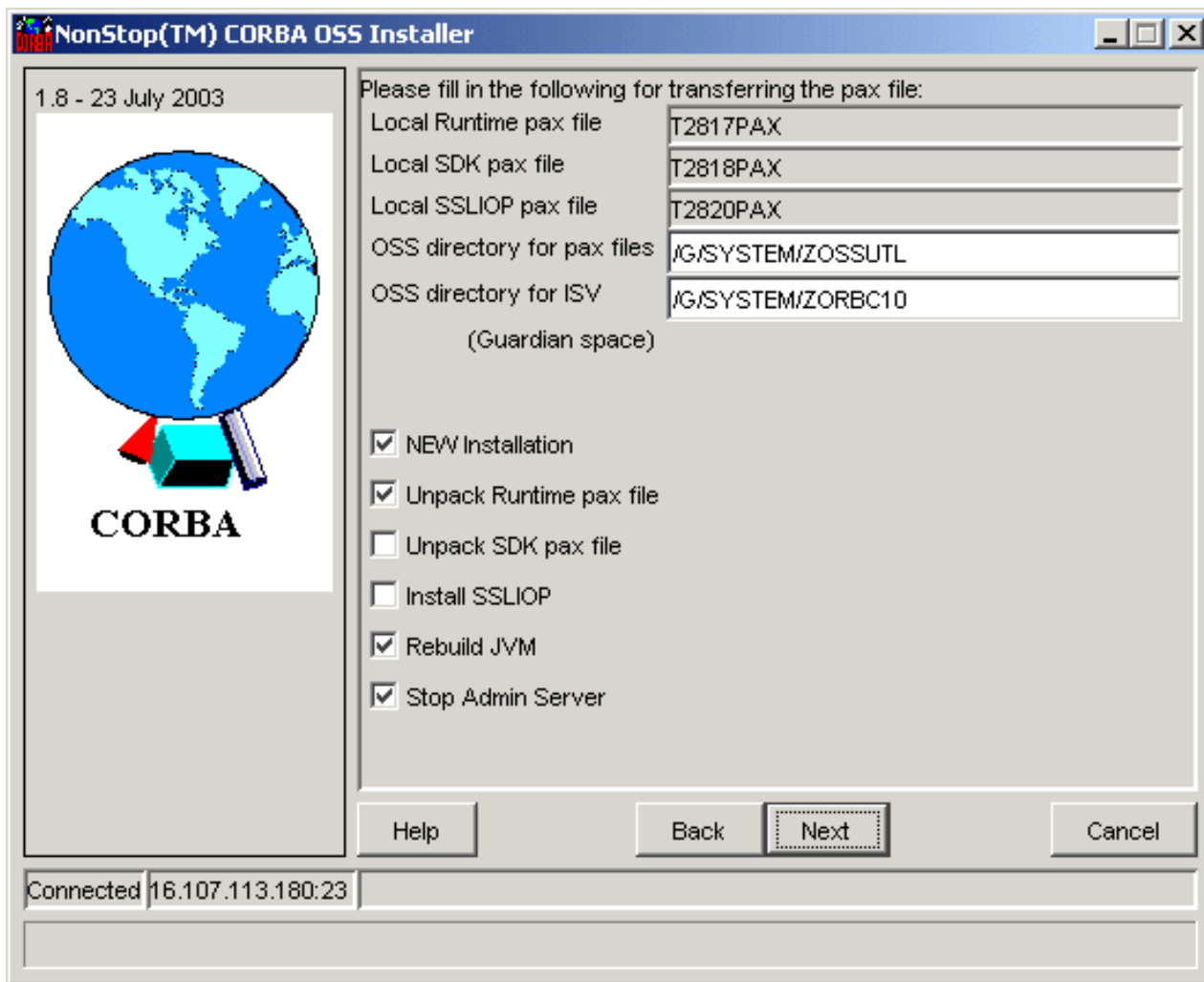
4. Supply a value for the OSS directory for pax file field. This value should be the installation subvolume (ISV) where IPSetup placed the software. You may type a new value as needed. Review the check boxes. Click **Next** to continue.

**i Note:** You should leave the box checked for rebuilding the JVM.

**i Note:** If you used DSM/SCM, the PAX files are already unpacked, and you should uncheck the box for unboxing the files. If you did not use DSM/SCM, leave the box checked.

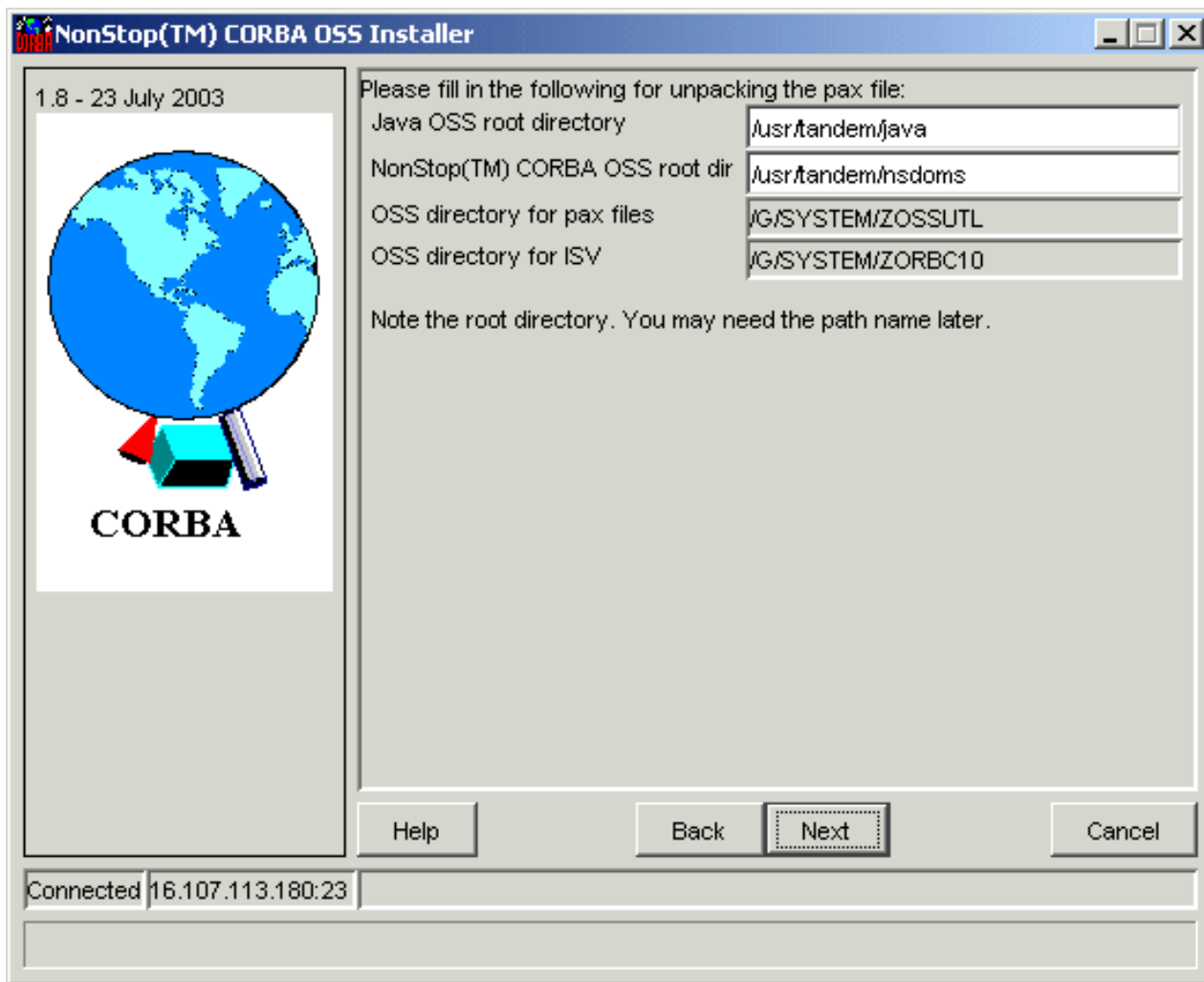
Field Name	Default Value	Description
Local Runtime pax file	T2817PAX	Cannot be changed.
Local SDK pax file	T2818PAX	Cannot be changed.
Local SSLIOP pax file	T2820PAX	Cannot be changed.
OSS directory for pax files	/G/SYSTEM/ZOSSUTL	Must be unique across all installed products. The location where the pax and other installation files are copied. Make a note of this value: it is used when bringing up the NonStop CORBA services.

OSS directory for ISV	G/SYSTEM/ZORBC11	The installation subvolume
NEW installation	checked	<p>Uncheck this box only if you are updating an existing NonStop CORBA 2.6.1 installation. When updating a previous CORBA 2.6 installation, program files are copied to the system but existing configuration files are untouched. Leaving this box checked means both program and configuration files (if they exist) are overwritten.</p> <p>Leave this box checked to unpax the Runtime. Uncheck this box if DSM/SCM has already unpaxed the file. Uncheck this box if a previous NonStop CORBA installation was interrupted after the pax files were unpaxed.</p>
Unpack Runtime pax file	checked	<p>Check this box to unpax the SDK. Uncheck if DSM/SCM has already unpaxed the file. Uncheck this box if a previous NonStop CORBA installation was interrupted after the pax files were unpaxed.</p>
Install SSLIOP	unchecked	<p>Check this box to install SSLIOP.</p> <p>The NonStop CORBA native library contains Guardian information and is linked into the JVM. Uncheck this box only if the native library has not changed since the last NonStop CORBA installation.</p>
Rebuild JVM	checked	<p>Leaving this box checked means the installer will stop the Admin Server before the installation process begins. Uncheck this box only if you don't want to stop the Admin Server. When the installation is completed, the installer will attempt to start the Admin Server.</p>
Stop Admin Server	checked	



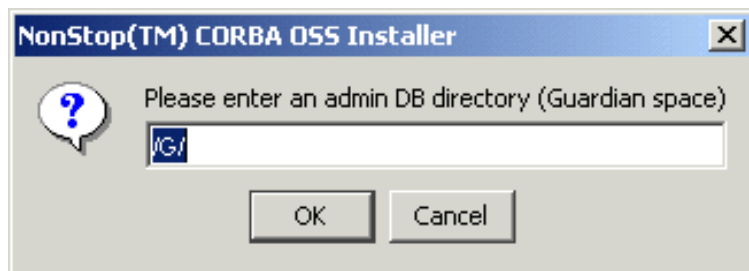
5. Supply values for the Java OSS root directory and the NonStop CORBA OSS root dir fields by accepting the defaults or typing new values. Click **Next** to continue.

Field Name	Default Value	Description
Java OSS root directory	/usr/tandem/java	Must be unique across all installed products. The location of NonStop Server for Java. If you use DSM/SCM, you may not change this value.
NonStop CORBA OSS root dir	usr/tandem/nsdoms	Must be unique across all installed products. The location of NonStop CORBA files. If you use DSM/SCM, you may not change this value.
OSS directory for pax files	Displays value entered previously.	The location where the pax and other installation files are copied.
OSS directory for ISV	Displays value entered previously.	The location of the installation subvolume.



6. You will be prompted if to install EMS templates at this point. If you wish to install EMS templates, click **YES**. If you click YES, you will be prompted again whether you want to allow the EMS template installation to run in the background.
7. Specify an admin DB directory. An admin dB directory stores configuration information, such as the list of security domains and the business domains associated with each security domain.

- If you indicated in [Step 4](#) that this installation is a new installation, you are prompted to define an admin dB directory. Type the directory path of the admin dB directory and click **OK**. The format is:  
*/G/volume/subvolume/filename*

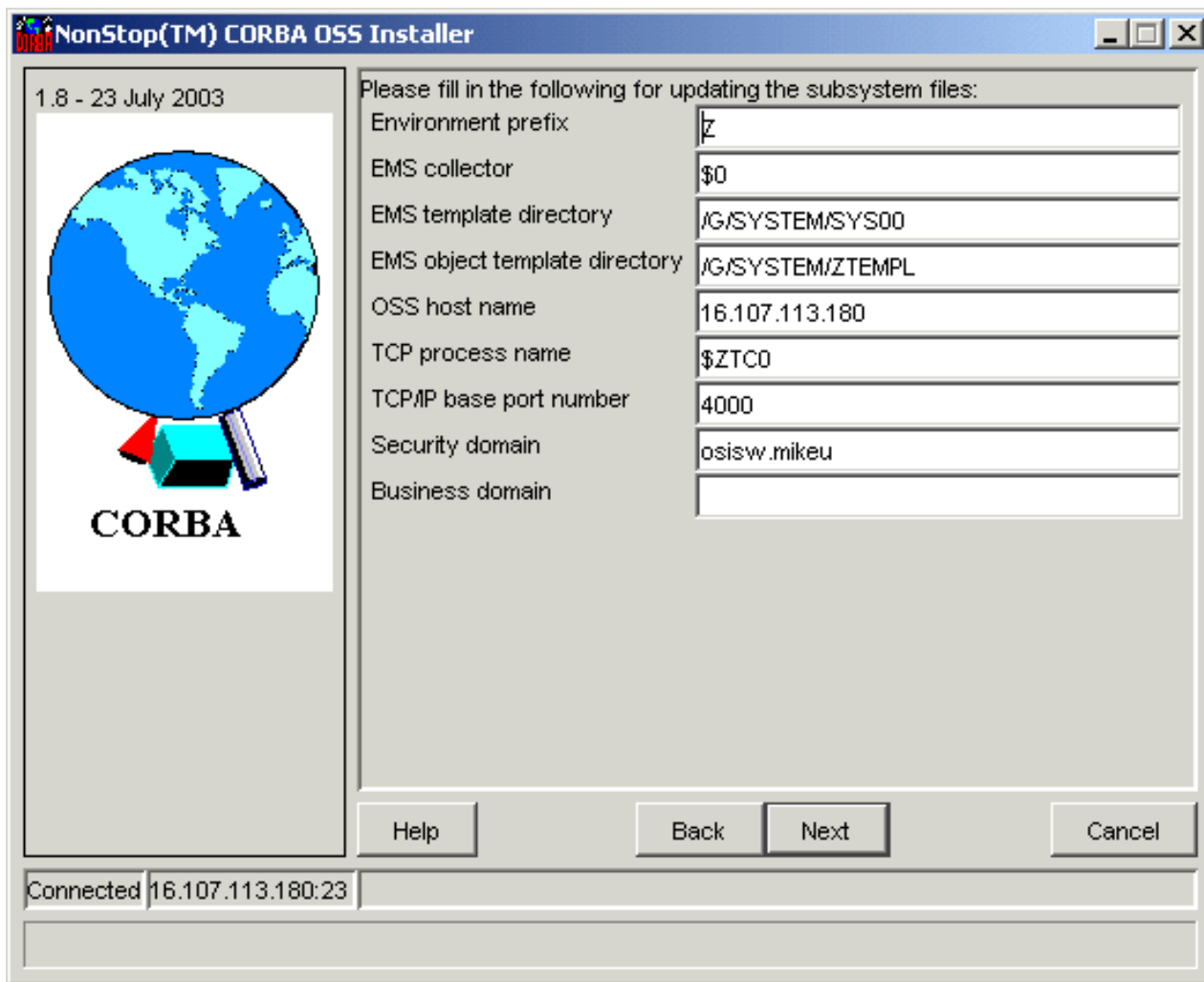


- If you indicated in [Step 4](#) that this installation is an update to a previous installation, you are prompted to select your admin dB directory. Select your directory from the list, then click **OK**. Normally you will have only one directory.
8. If you indicated in [Step 4](#) that this installation is an update to a previous installation, go to [Step 9](#). If you indicated this is a new installation, fill in the next group of fields.

Supply values for the Environment prefix, EMS collector, EMS template directory, EMS

object template directory, TCP process name, TCP/IP base port number, Security domain, and Business domain fields by accepting the defaults or typing new values. Click **Next** to continue.

<b>Field Name</b>	<b>Default Value</b>	<b>Description</b>
Environment prefix	Z	Must be unique across all installed products. One letter attached to the front of some NonStop CORBA process names and some NonStop Services for CORBA process names.
EMS collector	\$0	Name of the EMS collector.
EMS template directory	/G/SYSTEM/SYS00	Location of the EMS Templates.
EMS object template directory	/G/SYSTEM/ZTEMPL	Location of the EMS template object files.
OSS host name	Displays value entered previously.	Name or IP address of the host system.
TCP process name	\$ZTC0	TCP/IP process name for the host system. If the system has multiple TCP/IP processes, designate a specific process here for IIOP communications.
TCP/IP base port number	4000	Must be unique across all installed products. Port number for the host system, used for passive IIOP communications. The base port number and several subsequent port numbers are already in use.
Security domain	OSS user ID	Contains the processes and configuration data manageable under a user name and password. A security domain name can contain alphabetic, numeric, and underscore characters. It does not have to be of the <i>x.y</i> format like the OSS user ID. Example: finance.
Business domain	None. A business domain name is required for the installation to continue.	Contains a collection of related business application processes and configuration data. A business domain name can contain alphabetic, numeric, and underscore characters. Example: payroll.



9. The NonStop CORBA OSS Installer wizard indicates that the installation is complete. Click **Finish**.
10. If you did not choose to install the EMS template files using the GUI installer, then you should install them now by using the `INSTALL^TEMPLATES TACL` macro. The `INSTALL^TEMPLATES TACL` macro is distributed on the SUT and can be found in `volume.ZINSAIDS.TACLMACS`.

## Install the NonStop Distributed Component Console on the PC

The console installer installs the HP NonStop Distributed Component Console software and the Java Runtime Environment (JRE) on your PC. You have the option to not install the JRE if you already have it installed.

1. Double-click on the icon for the Console Installer file (`WS_SW\NDSCONSOLE.EXE`) on your CD to start the installation. Click **Next** when the welcome screen appears.
2. On the Choose Destination Location screen, click **Next** to accept the default location or browse to another location.
3. The Java Runtime Location screen has a check box to choose installing the Java Runtime Environment (JRE). Check the box if you do not already have the JRE installed. If you already have the JRE installed, check the directory shown to see if it is correct. If not, click the browse button to select the correct directory. Click **Next** to proceed.
4. On the Select a Program Manager Group screen, click **Next** to accept the default or choose another group. On this screen you can also choose to have a program icon placed on your desktop.
5. On the System Modifications screen, leave the Modify Java installation check box checked unless you wish to

manually modify your Java installation. Click **Next** to continue.

6. On the Start Installation screen, click **Next** to proceed with the installation.
7. If you have chosen to install the JRE in Step 3, the Java Runtime Environment Setup program will take over and you will be asked to agree to the Java licensing agreement. Click **Yes** to continue. As the installation runs, the installation progress is displayed on the Installing screen. When the JRE installation is complete you will be returned to the NonStop Distributed Component Console Installer program.
8. Click **Finish** to complete the installation.

See [Start NonStop CORBA Server Processes on the Host System](#) to run the console.

This completes the installation of NonStop CORBA 2.6.1. At this point all tools and services are available to run except `esadmin`, the GUI-based Event Service management tool. If you want to use `esadmin`, you must install JORB Lite as described in the following section. The `esadmin` tool is described in the *NonStop CORBA 2.6.1 Programmer's Guide for Java*. You do not need to use `esadmin` to manage the Event Service; you can use the NonStop Distributed Component Console instead.

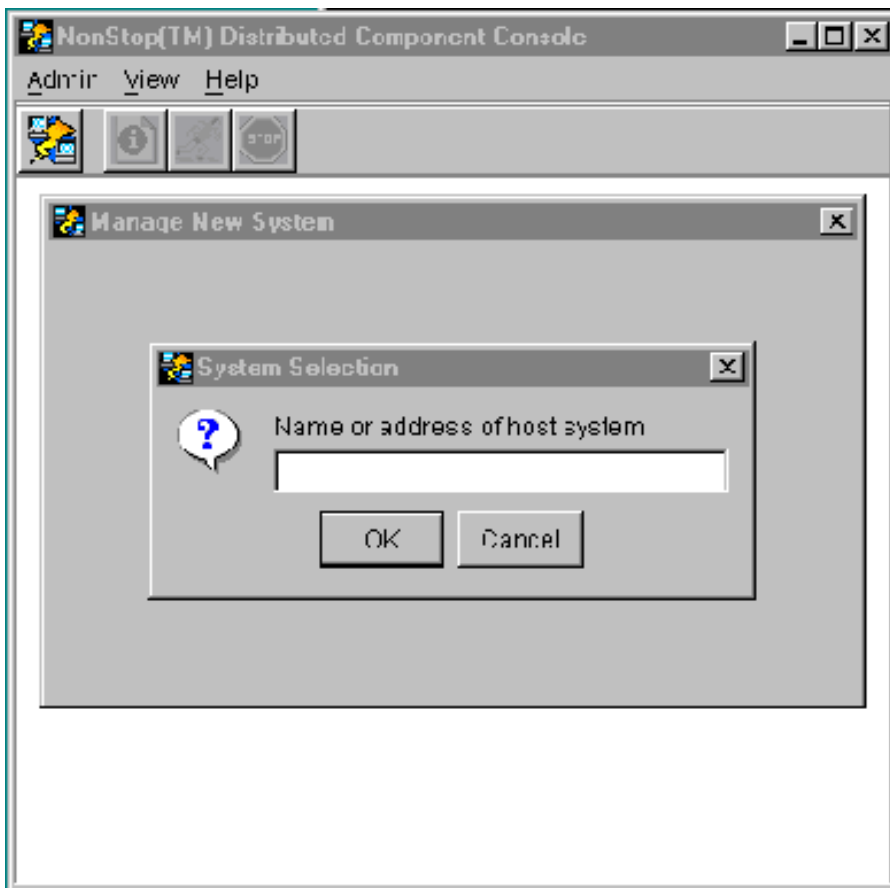
## Install JORB Lite on the PC

If you write applications in Java you may find JORB Lite useful. If you write applications only in C++ you do not need to install JORB Lite on the PC. See [Installing JORB Lite](#) and [Running JORB Lite](#) for more information.

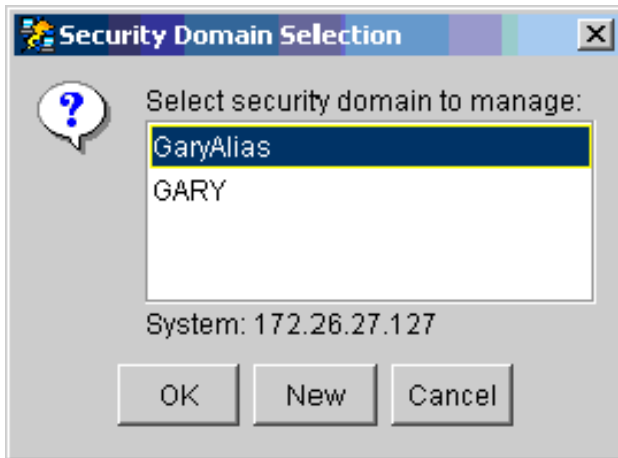
## Start NonStop CORBA Server Processes on the Host System

Use the NonStop Distributed Component Console to start the NonStop CORBA server processes and to verify their status. Alternatively you can log on to the OSS environment and use the `nsdstart` script. That method is discussed under [Alternative Installation Method](#).

1. Launch the Console on the PC by double-clicking on the NonStop Console icon placed on the desktop by the installer.  
The main Console window appears.
2. Select **Manage System** from the Admin menu or click **Manage a system** (leftmost button) on the tool bar. The System Selection dialog appears.
3. You may type either the name or the IP address of the host system. If you enter a name, it must be one that can be found by a host name lookup (usually performed by the Domain Name Service). Then click **OK**. The Security Domain Selection dialog appears.

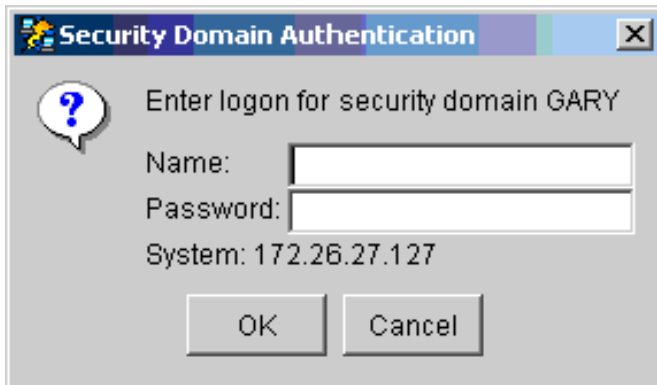


4. Select the security domain you defined during the installation, then click **OK**



The Security Domain Authentication dialog appears.

5. Log on to the security domain by typing the user name and password for the security domain, then click **OK**.

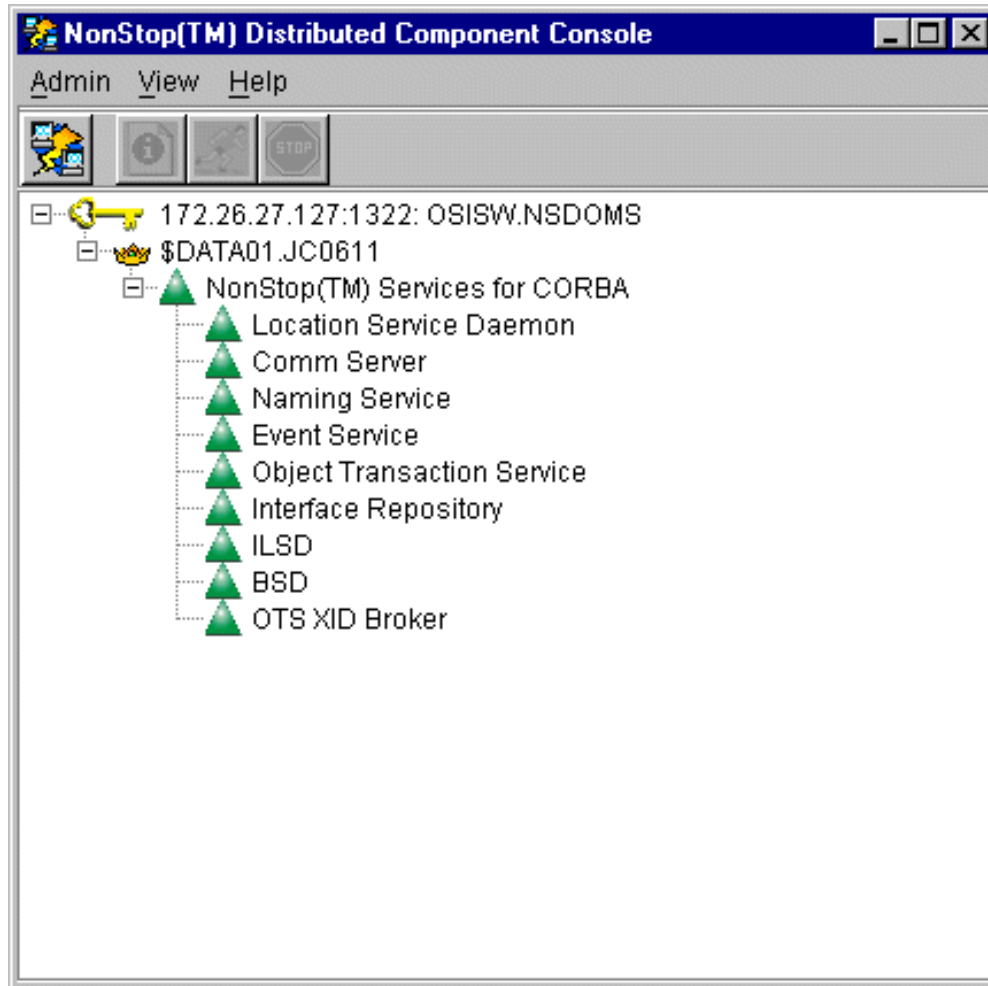


The Business Domain Selection dialog appears.

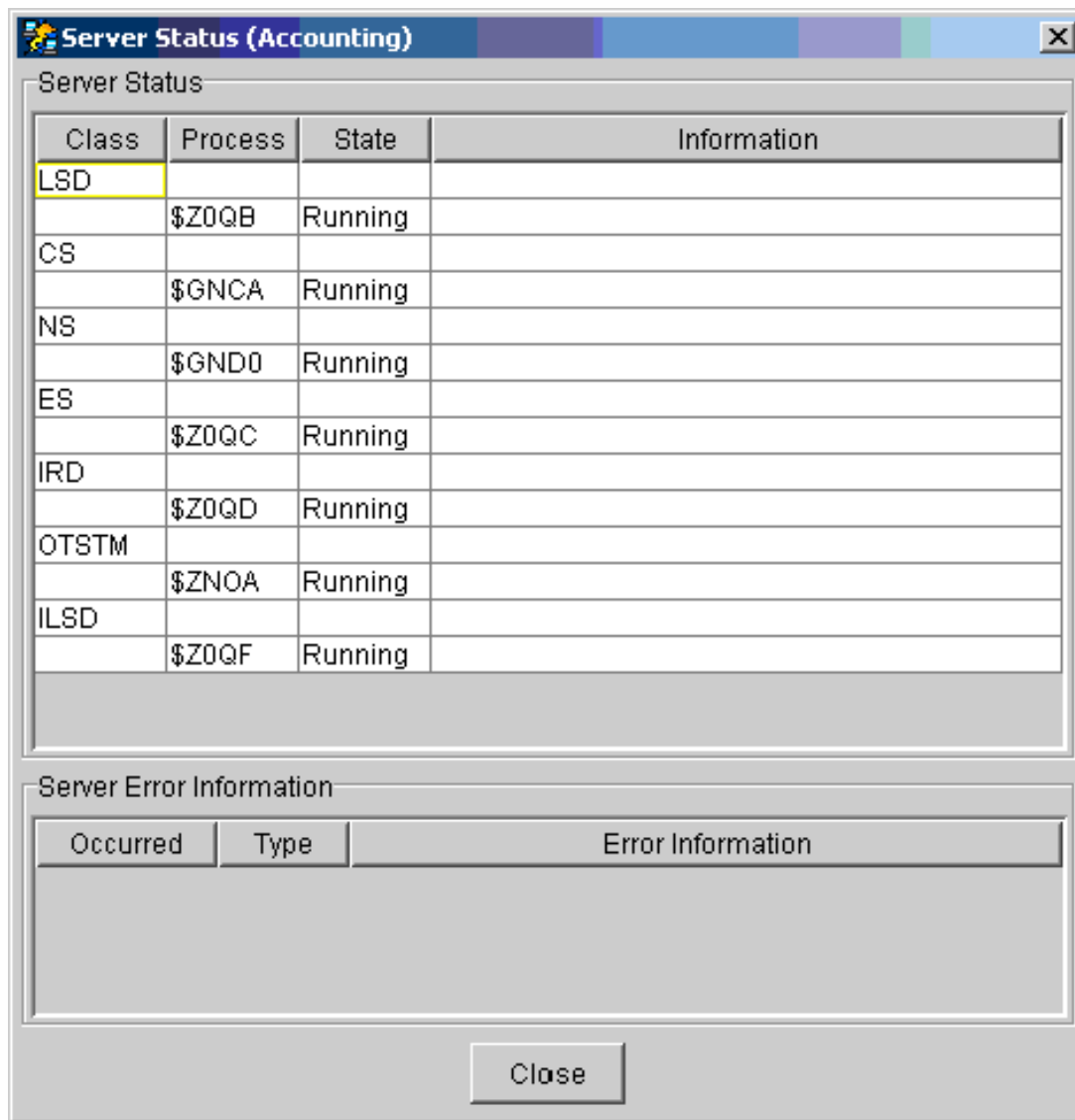
**i Note:** You can define additional security and business domains at a later time. See the Console online help for instructions.

6. Select the business domain you defined during the installation, then click **OK**. The business domain appears on the main Console window.
7. Click on the plus sign (+) next to the business domain to show the NonStop Services for CORBA node under the business domain. Click on the plus sign (+) next to the node to expand the list to see the services.
8. Select **NonStop Services for CORBA** and click **Start Running** (second button from the right) on the tool bar. All NonStop CORBA server processes are now started under a PATHMON process.

The status indicator next to each server process should be a green triangle that points up.



9. Verify the status of the server processes by doing the following:
  - a. Right-click **NonStop Services for CORBA**.
  - b. Select **Server Status** from the pull-down menu. A detailed status of each server is displayed.



## Modifying an Existing NonStop CORBA 2.6.1 Installation to Add the IIOP/SSL features

If you previously installed NonStop CORBA 2.6.1, but did not install the optional IIOP/SSL features, you can add those features. To check whether IIOP/SSL is already installed, run the `stack` sample program configured for IIOP/SSL as described in [Using the Stack Example with IIOP/SSL](#)

### Installing IIOP/SSL on an Existing NonStop CORBA 2.6.1 System

1. If CORBA is running, stop it by using the Console or the `nsdstop` script.
2. Run the NonStop CORBA OSS Installer GUI or the `install` script, just as you did for the initial installation of NonStop CORBA 2.6.1. If you are running the installer only to add the IIOP/SSL features, choose to:
  - Uncheck the NEW Installation box
  - Uncheck the Unpax Runtime pax file box
  - Leave the Unpax SDK pax file box unchecked
  - Check the Install SSLIOP box
  - Uncheck the Rebuild JVM box

- Leave the Stop Admin Server box checked.
3. Click **Next** to continue, and follow the prompts to complete the installation.
  4. Verify that IIOP/SSL is installed by running the `stack` sample program configured for IIOP/SSL as described in [Using the Stack Example with IIOP/SSL](#).

## Alternative Installation Method

You may choose to install NonStop CORBA on the NonStop system by using the Open System Services (OSS) shell prompt. Before starting the installation be sure to follow the instructions given in [Before You Begin the Installation](#) and have run the IPSetup utility as described in [Run the IPSetup Program](#). Optionally, you may have run the DSM/SCM planner interface as mentioned in [Run the DSM/SCM Planner Interface](#). To use the alternative method, you run the `install` script.

### Run the Install Script

Start the OSS shell by typing:

```
>osh
```

Move to the directory where you transferred the OSSPAX file:

```
>cd /G/SYSTEM/ZORBC11
```

Start the script by typing

```
>install
```

Follow the script prompts to completion. If the script encounters an error, it will back up to the closest logical starting place and allow you to re-enter information.

When the script prompts you to unpack the pax files, answer N if you have used DSM/SCM to place the files on your system and unpatch them. Answer Y if you have not used DSM/SCM.

After you have completed the installation, you can start the NonStop Services for CORBA by running the NonStop Distributed Console as described in [Start NonStop CORBA Server Processes on the Host System](#). Alternatively, you may start by logging on to the OSS environment and running the `nsdstart` shell script. See the *NonStop CORBA 2.6.1 Administration Guide* for more information about using scripts to start and configure your system.

### Sample Script Run

A sample run of the script follows. In this example default values are accepted wherever they are given.

```
osh prompt> install
```

```
=====
```

```
START NonStop(TM) CORBA INSTALLATION
installer version: main/tools/installer/install#33 , 2003/06/13
insthlpr version : main/tools/installer/insthlpr#48 2003/06/03
```

```
=====
```

```
Stop Admin Server (y/n) [default Y] ?:
```

```
Is this a new installation (y/n) [default Y] ?:
```

Unpack the pax files (y/n) [default Y] ?:

**Pax file**

=====

The pax file to unpack.

ENTER the pax file or leave blank to use the current  
value G/SYSTEM/ZOSSUTL/T2818PAX :

**Runtime pax file**

=====

The runtime pax file to unpack.

ENTER the runtime pax file or leave blank to use the current  
value G/SYSTEM/ZOSSUTL/T2817PAX :

Install SSLIOP (y/n) [default N] ?: Y

**SSLIOP pax file**

=====

The SSLIOP pax file to unpack

ENTER the IIOP/SSL pax file or leave Blank to use the current  
value /G/SYSTEM/ZOSSUTL/T2820PAX :

**JAVA\_HOME directory**

The root OSS directory for your JAVA installation.

ENTER the directory here or leave blank to use the current  
value /usr/tandem/Java :

**OSS Installation directory**

The root OSS directory for your installation.

ENTER the directory here or leave blank to use the current  
value /usr/tandem/nsdoms :

existing admin dB directories are:

*list of dB directories*

**Admin dB directory**

The root Guardian directory for the admin dB

ENTER the directory here (NEW or /G/vol/subvol): /G/system/master

**Prefix letter**

Specifies a unique prefix letter for the subsystem process names.  
Use this environment variable to prevent process-name conflicts if  
you have multiple NonStop CORBA installations on a single system.

ENTER prefix letter here or leave blank to use the current  
value Z :

=== CHECKING for existing process \$ZNDM ===

### **Collector Process**

Specifies the EMS collector process used by the subsystem. The name is defined in NonStop Kernel format (for example, \$ZNSD0).

ENTER the Collector process you want to use or leave blank to use the current value \$0:

### **EMS template directory**

The location of the system file TEMPLATE is required for configuring and creating the template files. If you have multiple copies of this file you should use the one that is open.

ENTER either a new Guardian location or leave blank to use the current value /G/SYSTEM/SYS00 :

### **EMS object template directory**

The destination location for EMS TEMPLATE object files is required for configuring and creating the new template files.

ENTER either a new Guardian location or leave blank to use the current value /G/SYSTEM/ZTEMPL :

### **Host name of machine**

The Name of host machine on which product runs. You can use either the symbolic name, the actual IP address, or the full symbolic name of the host.

ENTER host name here  
value : 155.154.153.152

### **TCP process name**

Specifies the TCP process name.

ENTER the TCP process name here or leave blank to use the current value \$ZTC0 :

=== CHECKING for existing process \$ZTC0 ===

### **TCP/IP base port number**

The port number you specify here and the next three sequential ports will be allocated for the subsystem.

ENTER base port number or leave blank to use the current value 4000 :

### **Security domain**

Specifies the security domain name.

ENTER security domain  
value USER.NSDOMS:

## Business domain

Specifies the business domain name.

```
ENTER business domain
value : bus
```

```
Do you want to build EMS templates (y/n) [default Y] ?:
```

```
Rebuild JVM (y/n) [default Y] ?:
```

## CURRENT SETTINGS

```
Guardian root directory : /G/system/zorbsdk
Java OSS root directory : /usr/tandem/java
OSS root directory : /usr/tandem/nsdoms
Admin dB root directory : /G/SYSTEM/MASTER
PAX file : G/SYSTEM/ZOSSUTL/T2818PAX
RT PAX file : G/SYSTEM/ZOSSUTL/T2817PAX
SSLIOP PAX file : G/SYSTEM/ZOSSUTL/T2820PAX
Environment prefix : Z
EMS collector : $0
EMS template directory : /G/SYSTEM/SYS00
EMS template object dir: /G/SYSTEM/ZTEMPL
OSS host name : 155.154.153.152
TCP process name : $ZTC0
TCP/IP base port number : 4000
Security domain : USER.NSDOMS
Business domain : bus
```

```
starting to unpack files
USTAR format archive
list of unpaxed files
```

```
updating installed files
```

```
building config dB
```

```
initializing LSD
```

```
initializing Name Service
```

```
building EMS templates (in background)
check /usr/tandem/nsdoms/log/bldems.out for details
```

```
updating JVM
```

```
starting Admin server
```

```
=====
NonStop(TM) CORBA INSTALLATION COMPLETE
```

=====

# Installing Multiple Copies of NonStop CORBA 2.6.1 on the Same Server

**i** **Note:** DSM/SCM allows only one location for each Independent Product installation. If you want to install a second copy of NonStop CORBA, you must do the installation directly, without using DSM/SCM. The DSM/SCM database will be unaware of any such subsequent installations.

For second or subsequent copies of NonStop CORBA installed on the same system,

- Each installation should be considered a new installation.
- Each installation must be in its own directory.
- The prefixes must be unique.
- Port numbers must be unique. NonStop CORBA assigns port numbers for the AdminServer, Comm Server (TCP/IP), BSD, ILSD, and LSD during installation.
- Each installation will have its own NSDCFGDB and NAMINGDB.

---

[<< previous](#)

Chapter 1. Introduction to NonStop CORBA

[up ^](#)

[home](#)

[next >>](#)

Chapter 3. Considerations for Migrating from Previous Releases

# Chapter 3. Considerations for Migrating from Previous Releases

## [Product Level Migration Information](#)

### [Overview](#)

### [Upgrading From NonStop CORBA 2.3 to NonStop CORBA 2.6.1](#)

### [Moving Applications from NonStop CORBA 2.3 Update 3 or Later to NonStop CORBA 2.6.1](#)

## Product Level Migration Information

### Overview

NonStop CORBA 2.6.1 is compatible with prior versions of NonStop CORBA. New features have been added, but no features have been removed. Some general considerations follow:

- The GUI installer scripts for the SDK and RTK have been combined into one script, with choices for installation of the SDK and of SSL as options.
- The installation script has been enhanced to allow installation of SSL as an option.
- DSM/SCM support has been added to help manage installation of files in the OSS environment.
- The Console has been altered. Console installations from previous versions should be uninstalled, and the new Console should be installed.
- NonStop CORBA 2.6.1 can be managed with the same management tools used in previous versions.
- NonStop CORBA 2.6.1 should perform as well or better than earlier versions.

## Upgrading From NonStop CORBA 2.3 to NonStop CORBA 2.6.1

You should always install NonStop CORBA as a new installation, even if you have a previous installation of NonStop CORBA 2.3. If you use the GUI installer, follow the procedures in [Installing NonStop CORBA](#), and leave the **NEW installation** check box checked. If you use the installation script as described in [Alternative Installation Method](#), leave the default for the question *Is this a new installation*.

Installing NonStop CORBA 2.6.1 as a new installation causes the previous settings and databases to be

overwritten with new CORBA 2.6 settings and databases (for example, NSDCFGDB, NAMINGDB, nsdom.ir, env.sh).

# Moving Applications from NonStop CORBA 2.3 Update 3 or Later to NonStop CORBA 2.6.1

For all applications,

1. IDL files must be recompiled by the CORBA 2.6 IDL compiler.
2. Java source files must be recompiled using NonStop Server for Java version 3.1.1.
3. Be aware that the default port for the ILSD has changed. It is now set to *\$portnumber3* at installation.

Some applications may need to be changed, depending on whether they use the constructs discussed in the following paragraphs.

- The value of the property `org.omg.CORBA.ORBSingletonClass` is `com.tandem.nsdomb.ORB.ORB_ORB_Singleton`. (In the past, the value was `com.tandem.nsdomb.ORB.ORB_ORB`.) If an application has been setting this property explicitly, modify the application.
- `org.omg.CORBA.ORB.init()` returns a singleton ORB, whose functionality is limited (according to OMG specifications). (In the past, `org.omg.CORBA.ORB.init()` [`singleton ORB`] and `org.omg.CORBA.ORB.init(String[], Properties)` both returned an ORB.) If an application has been using a singleton ORB where a non-singleton ORB should have been used, modify the application.
- For the ORB to process incoming GIOP requests, the server application must specifically call `org.omg.CORBA.ORB.run()` or a similar method. (In the past, an application did not have to call `org.omg.CORBA.ORB.run()` or a similar method in a server-specific application.)
- The default GIOP version is 1.2 (previously, it was 1.0). If an application requires GIOP 1.0, modify the application's profile to add the key/value `giop_version 1.0`.
- For `corbaloc` and `corbaname` the proper syntax is `::host`, according to OMG specifications. (In the past, the `corbaloc` and `corbaname` syntax was `://host`.)
- If JTS is not being used and `jts.jar` is included in the application's classpath, application performance can be impacted. Remove `jts.jar` from the application's classpath if JTS is not being used. (In the past, including `jts.jar` in the application's classpath when JTS was not being used had little impact on application performance.)
- The adapter activator must extend `org.omg.PortableServer._AdapterActivatorLocalBase`. (In the past, an application adapter activator extended `org.omg.PortableServer.AdapterActivatorPOA`.)

- The servant activator must extend `org.omg.PortableServer._ServantActivatorLocalBase`. (In the past, an application servant activator extended `org.omg.PortableServer.ServantActivatorPOA`.)
- The servant locator must extend `org.omg.PortableServer._ServantLocatorLocalBase`. (In the past, an application servant locator extended `org.omg.PortableServer.ServantLocatorPOA`.)
- The servant manager needs to extend `org.omg.PortableServer._ServantManagerLocalBase`. (In the past, an application servant manager extended `org.omg.PortableServer.ServantManagerPOA`.)
- Some objects created by the ORB extend `org.omg.CORBA.LocalObject`, which, in turn, extends `org.omg.CORBA.Object`, whose `_release()` method throws `org.omg.CORBA.NO_IMPLEMENT` exception, if invoked. (In the past, calling the method `_release()` on a CORBA object always worked.)
- The nonstandard helper class `org.omg.PortableServer.PortableServer`, which contained methods `string_to_ObjectId` and `ObjectId_to_string`, is unavailable. (In the past, this helper class was available.) The method `string_to_ObjectId(String value)` may be replaced by `value.getBytes()`. The method `ObjectId_to_string(byte[] value)` may be replaced by `new String(value)`. Refer to the Java documentation for issues relating to these suggested replacements.
- The Java ORB and JORB Lite implement standard Portable Interceptors (PIs). (In the past, the Java ORB and JORB Lite implemented pre-standard PIs.) Note that message interceptors are unavailable with standard PIs. If your application used the pre-standard PIs, you need to re-implement your PIs to the standard PIs.
- Generated narrow methods in Helper classes will throw an `org.omg.CORBA.BAD_PARAM` exception if the narrow fails. (In the past, generated narrow methods in Helper classes did not throw an `org.omg.CORBA.BAD_PARAM` exception.)

---

[<< previous](#)

[up ^](#)

[next >>](#)

[Chapter 2. Installing NonStop CORBA](#)

[home](#)

[Chapter 4. Running the Java Stack Example](#)

# Chapter 4. Running the Java Stack Example

[The Stack Example Overview](#)

[Building the Stack Sample](#)

[Configuring NonStop CORBA for the Stack Example](#)

[Running the Stack Example on OSS](#)

Once you have installed and configured your NonStop CORBA software, you can test the operation of your system by running a simple CORBA program.

The Stack example is a simple but fully functional CORBA program that has both a client and a server implemented in C++ and Java. This section discusses the Java version of the example. Because it contains both client and server applications, the Stack example is a perfect test for your newly installed NonStop CORBA system.

## The Stack Example Overview

The Stack example resides in the `javasamples/stack` subdirectory of your NonStop CORBA installation. While the intent of this topic is to test your installation, the `README` file in the Stack directory describes the workings of the program files.

The Stack example is a program that implements both a CORBA client and server in Java. It is a complete and working program that demonstrates:

- An object interface definition written in the Interface Definition Language (IDL)
- A POA server that embodies the implementation of the interface defined by the interface definition
- Two POA servers that embody the implementation of the interface defined by the interface definition
- A client application which makes use of an instance of a stack object

In addition to these items, the Stack example can be used to demonstrate both portability and interoperability. To demonstrate portability, you can move this sample to another platform running an implementation of CORBA. To demonstrate interoperability, you can build and run the client application on one platform, and run it with the server application built on another platform.

# Building the Stack Sample

The Stack example contains a file named `Makefile`, which builds both the client and the server applications. The `Makefile` performs the required steps to construct the executable components.

To build the sample:

1. Log on to the system and go to the directory `javasamples/stack`.
2. Verify that the `etc/env.sh` script has been sourced and that the environment variable `JAVA_HOME` points to the installation of the NonStop Server for Java.
3. Run the `Makefile` in the `javasamples/stack` directory.

The `Makefile`:

- Uses the NonStop CORBA IDL compiler to process the interface contained in `stack.idl` file
- Builds the server executable by using the NonStop Server for Java compiler to compile its components
- Builds the client executable by using the NonStop Server for Java compiler to compile its components

The NonStop CORBA IDL compiler `nsdidl` takes the `stack.idl` file as input and produces the following 12 files:

- `Stack.java`
- `StackHelper.java`
- `StackHolder.java`
- `StackOperations.java`
- `StackPOA.java`
- `_StackStub.java`
- `StackPackage/STACK_OVERFLOW.java`
- `StackPackage/STACK_OVERFLOWHelper.java`
- `StackPackage/STACK_OVERFLOWHolder.java`
- `StackPackage/STACK_UNDERFLOW.java`
- `StackPackage/STACK_UNDERFLOWHelper.java`
- `StackPackage/STACK_UNDERFLOWHolder.java`

These files contain the CORBA-specified Java language-mapping translation of the interface definition provided in the `stack.idl` file. The compilation steps use the NonStop Server for Java tools to

compile the programs. During the compilation, the Java compiler uses the NonStop CORBA implementation files in the `$NSD_ROOT/lib/jorb.jar` file.

## Configuring NonStop CORBA for the Stack Example

Before you can run the Stack example, you must update the configuration database by adding an entity and several key-value pairs, but only after you have installed and configured NonStop CORBA.

To update the configuration database for the Stack example, add the `tcp_server` entity to the NonStop CORBA configuration database by entering:

```
entityaddkeyvalue tcp_server@ORB tcp_server true
entityaddkeyvalue tcp_server@ORB port_number 0
entityaddkeyvalue tcp_server@ORB host_name nnn.nnn.nnn.nnn
```

Note that in setting the host name you can enter either a host name or an IP address.

## Running the Stack Example on OSS

The Stack example is now ready to run. If your NonStop CORBA system is not already running, you must start it by using the Console or the `nsdstart` script.

To run the Stack example, open two OSS windows. In each window, navigate to the directory where the Stack example was built, `javasamples/stack`. Also, make sure that each window has the correct environment setup, which is obtained by sourcing in the `env.sh` script.

In one window, start the server program by using the following command:

```
java Server -ORBprofile tcp_server
```

In the other window, start the client program by using the following command:

```
java Client -ORBprofile tcp_client
```

When you run the client, you will see the following output in its window:

```
Top: 100
Top: 200
Top: 300
Top: 400
Top: 500
Top: 600
Top: 700
Top: 800
Top: 900
Top: 1000
Got STACK_OVERFLOW exception as expected.
Pop: 1000
```

Pop: 900  
Pop: 800  
Pop: 700  
Pop: 600  
Pop: 500  
Pop: 400  
Pop: 300  
Pop: 200  
Pop: 100

Got `STACK_UNDERFLOW` exception as expected.

If you see the output as described above, you have successfully built and run the Stack example on your NonStop CORBA system.

Once you are satisfied with the output, you can stop the server program (in the server OSS window) by pressing `Ctrl-C`.

You can now begin developing your own CORBA clients and servers using NonStop CORBA system.

If you do not obtain the output as described above, either you have not correctly installed and configured NonStop CORBA, or you did not build and run the example program correctly. You can start the debugging process by using `PATHCOM` to view the status of your NonStop CORBA server-pool processes to see if the ORB processes are running.

[<< previous](#)

[up ^](#)

[next >>](#)

[Chapter 3. Considerations for Migrating from Previous Releases](#)

[home](#)

[Chapter 5. Installing JORB Lite](#)

# Chapter 5. Installing JORB Lite

[JORB Lite Overview](#)

[The JORB Lite Package](#)

[Installing JORB Lite](#)

[The JORB Lite Directory Structure](#)

[The PATH and CLASSPATH Environment Variables](#)

[Running the IDL Compiler](#)

## JORB Lite Overview

JORB Lite is a software development kit (SDK) and runtime environment that lets you develop CORBA applications in the Java programming language. Implemented in 100 percent pure Java, the JORB Lite SDK is a CORBA 2.6.1-based client and server development environment.

**i Note:** JORB Lite is suitable for development and test. It is not intended or supported for production use.

## The JORB Lite Package

JORB Lite contains the following files, tools, and sample programs:

The JORB Lite Package

Name	Description	Comments
<code>jorblite.jar</code>	JORB Lite classes	DO NOT UNJAR THIS FILE! This file must remain jarred for the Java compiler and Java interpreter to properly access the class files. This file contains all of the compile <code>.class</code> files for the SDK.

JORB Lite Tools

Name	Description	Comments
<code>idl2java.bat</code>	IDL compiler	Compiles an OMG IDL file into Java stubs and skeletons.

JORB Lite Examples

Name	Comments
Bank example	Located in the <code>javasamples/bank</code> directory.
Calc example	Located in the <code>javasamples/calc</code> directory
Event Service example	Located in the <code>javasamples/event_service</code> directory
Interoperable Naming Service example	Located in the <code>javasample/ins</code> directory
Jpathsend example	Located in the <code>javasamples/jpathsend</code> directory

JTS bank example	Located in the <code>javasamples/jts_bank</code> directory
Portable Interceptor example	Located in the <code>javasamples\portable-interceptor</code> directory
Stack example	Located in the <code>javasamples/stack</code> directory
Stack Object by Value example	Located in the <code>javasamples/stack_obv</code> directory
Stack Tie example	Located in the <code>javasamples/stack_tie</code> directory
Stock Quote example	Located in the <code>javasamples/stock</code> directory
Stock Dynamic example	Located in the <code>javasamples/stock_dynamic</code> example

## Documentation

Name	Comments
Product documentation	Located in the <code>docs\javapubs\javadocs\javadoc</code> directory

# Installing JORB Lite

JORB Lite resides in a self-extracting executable file on your NonStop CORBA compact disc. This installation procedure assumes you have the Java SDK (version 1.1.2 or newer) installed. To install JORB Lite:

1. From the product CD, run the `WORKSTATION\JORBLITE.EXE` file. You will be prompted for a directory where you want to place the files. The default directory is `c:\jorblite`. If you want to enter a different directory name do so and continue. The resulting [directory structure](#) is shown at the end of this installation procedure. Note that the installer will update your `PATH` and `CLASSPATH`.
2. Unzip the following documentation files in the same directory that they live in:  
`\docs\javapubs\javadocs\javadoc.zip`
3. To develop JORB Lite applications you must install a C language preprocessor such as `cl.exe` from Microsoft's Visual C++ (the preprocessor is needed by the IDL compiler).

You must ensure that the preprocessor program is accessible from the `PATH` environment variable, because the `idl2java` script executes IDL using the `cl.exe` program to preprocess information in the IDL file. If the IDL compiler cannot find the preprocessor, it will generate the error:

```
nsdidl: Error: nsdidl.PreprocessorException: failed to execute "cl -E -C filename.idl"
```

4. If you use Gnu's C preprocessor (`cpp`) you will need to edit `bin\idl2java.bat` to change "cl" to "cpp."
5. If you use the Java 2 platform you need to copy `lib\orb.properties` to your *Java 2 root directory*\jre\lib.
6. If you want to use `org.omg.CORBA.ORB.resolve_initial_references ("NameService")` you must also modify the `RootNamingContextIORFile` setting in the file `jorblite-dir\lib\lite.dbm` so that it looks like: `NS@name_service_settings`  
`RootNamingContextIORFile jorblite-dir\urls\NameService`
7. The file "NameService" must contain the root naming context IOR of the Naming Service running on a remote NonStop system. The NonStop CORBA subsystem creates this file when it is configured on the system. The file is in the directory: *NonStop CORBA installation dir*/urls on the NonStop system. You must copy this file from the NonStop system into the `jorblite-dir\urls` directory. You can use the NonStop Distributed Component Console or FTP to copy the file.

**i Note:** Copy this file to your PC after each NonStop CORBA reconfiguration on the NonStop system.

8. Modify `jorblite-dir\lib\lite.dbm` so that the `host_name` entry contains the host-name of your PC (the host from where you run JORB Lite).

When you unpack JORB Lite, it installs the file `env.bat` into the `etc` subdirectory. The installer also sets various environmental variables used by your JORB Lite setup. You may use the `env.bat` file to manually set the variables, which may need to be modified for your installation. Make sure that the following default environment variables set in this file refer to the correct directories on your PC system:

#### JORB Lite Environmental Variable Setting

Variable	Default Setting	Description
JORBLITE_HOME	<code>c:\jorblite</code>	Root directory of your JORB Lite installation.
JAVA_HOME	<code>c:\jdk 1.3.1</code>	Root directory of your JDK installation. Assumes that you have the 1.3.1 JDK installed.

For example, if you install JORB Lite into a directory other than `c:\jorblite`, you will have to modify the file `etc\env.bat` so that the `JORBLITE_HOME` environment variables reference the correct JORB Lite root directory.

## The JORB Lite Directory Structure

When you install JORB Lite, the installation process creates the following directory structure under your base JORB Lite directory:

#### The JORB Lite Directory Structure

```
+--bin
+--docs
|   +--javapubs
|       +--javadocs
+--etc
+--idl
+--jvasamples
|   +--bank
|       +--calc
|
|   +--event_service
|       +--ins
|
|   +--jpathsend
|       +--jts_bank
|
|   +--stack
|       +--stack-obv
|       +--stack-tie
|
|   +--stock
|       +--stock-dynamic
```

+--lib

## The PATH and CLASSPATH Environment Variables

The installer makes the following environment variable changes:

- The absolute path of the `\bin` directory is added to the `PATH` environment variable for your environment.
- A `CLASSPATH` environment variable is added and the absolute paths of `\lib\jorblite.jar` is added to this environment variable.

The `env.bat` files provided with JORB Lite set the `PATH` and `CLASSPATH` environment variables. If you change the system environment variables from those initially installed, make sure you remove the respective environment variable settings from the `env.bat` files contained in your JORB Lite installation.

## Running the IDL Compiler

The JORB Lite IDL compiler does not have a GUI interface you must run the compiler from the Windows command prompt. The IDL compiler uses the following syntax:

```
idl2java filename
```

In this command, *filename* is the name of the IDL file you want to compile.

---

[<< previous](#)

[up ^](#)

[next >>](#)

Chapter 4. Running the Java Stack Example

[home](#)

Chapter 6. Running JORB Lite

# Chapter 6. Running JORB Lite

[Building and Running the Stack Example](#)

[Building the Stack Example](#)

[Running the Stack Example](#)

[Calling NonStop CORBA Applications from JORB Lite](#)

[Running Clients and Servers on Remote Machines](#)

After you have installed JORB Lite, you can test your system setup by running one of the examples supplied with the JORB Lite package.

## Building and Running the Stack Example

To build and run the JORB Lite Stack example, open two DOS command-prompt windows. In each window, navigate to the `javasamples/stack` directory.

The Stack example includes several batch (`.BAT`) files to help you build and run of the Stack example:

Stack Example Files for JORB Lite

Name	Description
<code>make.bat</code>	This file generates the Java Language Mappings files from the <code>stack.idl</code> interface and builds the client applications and server applications.
<code>runclient.bat</code>	This file runs the client Java program.
<code>runserver.bat</code>	This file runs the server Java program.
<code>runserverpoa.bat</code>	This file runs the server Java program using a POA server.

In addition to these batch files, the Stack example also uses the file `lib\lite.dbm`, a text file that provides the configuration database for the JORB Lite environment.

## Building the Stack Example

Follow these steps to build the client and server applications:

1. Open a command-prompt window, and navigate to the JORB Lite Stack example directory.
2. Run the `make.bat` file to generate the Stack Java files and build the client and server applications.
3. Modify the `lib\lite.dbm` file by referring to your host name or IP address and a port number in the file.

## Running the Stack Example

You have now built the Stack client and server applications and are ready to test your JORB Lite system:

1. Open a second command-prompt window. In this window, navigate to the Stack example directory.
2. In the first command-prompt window, execute the Stack server program by running the `runserver.bat` file. (Note that you can also run the `runserverpoa.bat` file to run the POA implementation.)
3. From the second command-prompt window, execute the Stack client by running the `runclient.bat` file.

If you receive the following error, check that you have entered the values into the `lib\lite.dbm` file and that you have set up your environment correctly:

```
ERROR: Cannot find configuration database
Please verify that -ORBlitedbm <db> was specified in the argument to
```

The client should run and produce the following output:

```
Top: 100
Top: 200
Top: 300
Top: 400
Top: 500
Top: 600
Top: 700
Top: 800
Top: 900
Top: 1000
Got STACK_OVERFLOW exception as expected.
Pop: 1000
Pop: 900
Pop: 800
Pop: 700
Pop: 600
Pop: 500
Pop: 400
Pop: 300
Pop: 200
Pop: 100
Got STACK_UNDERFLOW exception as expected.
```

## Calling NonStop CORBA Applications from JORB Lite

The Java language allows you to "Write once, Run anywhere," which means that you can generate JORB Lite clients and servers and move their generated `.class` bytecode files to the NonStop Kernel for

execution in the NonStop CORBA environment. You can also move only the client or server `.class` and their associated IOR files and allow JORB Lite applications to interoperate with NonStop CORBA applications.

You can use JORB Lite as a build environment and NonStop CORBA as an execution environment. To test this technique, follow the steps above to [build the Stack example](#) on your PC. After building the example, you can move the generated `.class` files (using a program such as FTP) to a directory on your NonStop JORB OSS system.

To set up your NonStop CORBA application, follow the steps for configuring NonStop CORBA clients and servers as listed in the Administration Guide. You run the Stack example the same way you would if you had built it on your OSS system (see the Running the Stack Example for details).

This exercise shows that you can develop NonStop CORBA applications in a JORB Lite environment, move the compiled `.class` bytecode files to your OSS system, and run your Java CORBA applications within the NonStop CORBA environment.

## Running Clients and Servers on Remote Machines

To demonstrate CORBA interoperability between JORB Lite and NonStop CORBA, generate the JORB Lite clients and servers by following the steps in [Building and Running the Stack Example](#). Move the generated `.class` server files (using a program such as FTP) to a directory on your NonStop CORBA system.

Configure the Stack example as a server on your NonStop CORBA system, as outlined in the Administration Guide. Use the Configuration Tool to modify your NonStop CORBA configuration database so it recognizes your NonStop JORB server as a `tcp_server` server.

1. On your NonStop CORBA system, start an OSS window and source the script `etc/env.sh` to set up the environment.
2. Since the `.class` server bytecode files built using JORB Lite now reside on your NonStop CORBA system, you can execute the Stack application server by following the directions given in the Running the Stack Example topic.
3. When you run the Stack application server, it generates a stringified IOR file named `stack.ior` in the directory where the Stack server is executed. This file contains the stringified object reference to the Stack server object implementation executing under OSS.
4. Copy the `stack.ior` file into the directory containing the Stack client `.class` files on your PC by using a program such as FTP.
5. Run the client as described earlier in [Building and Running the Stack example](#). The stack client will read the `stack.ior` file and use the stringified IOR to locate the stack server executing under NonStop CORBA on the remote NonStop Kernel system, and you will see the same output as detailed [above](#).



---

# Index

## A

Application, migrating

NonStop CORBA 2.3 Update 3 to NonStop CORBA 2.6.1, [Moving Applications from NonStop CORBA 2.3 Update 3 or Later to NonStop CORBA 2.6.1](#)

Availability, [Availability and Fault Tolerance](#)

## B

Basic Object Adapter (BOA), [Portable Object Adapter](#)

Bootstrap Daemon (BSD), [Bootstrap Daemon \(BSD\)](#)

## C

Client/server computing, [Client/Server Computing with NonStop CORBA](#)

Comm Servers

definition of, [Network Session Concentration](#)

protocols, [Network Session Concentration](#)

CORBA compliance, [CORBA 2.6.1 Base](#)

## D

Data integrity, [Transaction Protection and Data Integrity](#)

## E

Error logging, [Error Logging and Tracing](#)

Event Service, [Event Service](#)

## F

Fault tolerance, [Availability and Fault Tolerance](#)

## H

Hardware and software requirements, [Hardware and Software Requirements](#)

## I

### IDL

compiler (NSDIDL), [IDL Compiler](#)  
interface, [Error Logging and Tracing](#)

### Installation

alternative method, [Alternative Installation Method](#)  
before you begin, [Before You Begin the Installation](#)  
Installer Setup Program, [Run the NonStop CORBA OSS Installer Setup Program](#)  
installer wizard, [Host Installation Installer Wizard](#)  
installing multiple versions of software, [Installing Multiple Copies of NonStop CORBA 2.6.1 on the Same Server](#)  
JORB Lite, [Installing JORB Lite](#)  
JORB Lite on PC, [Install JORB Lite on the PC](#)  
NonStop CORBA on host system, [Install NonStop CORBA on the Host System](#)  
NonStop Distributed Component Console on PC, [Install the NonStop Distributed Component Console on the PC](#)  
overview, [Installation Overview](#)  
performing, [Overview of Installation Steps](#)  
running the Install script, [Run the Install Script](#)  
subsystem resource cleanup, [Perform Subsystem Resource Cleanup](#)  
test JORB Lite install, [Running JORB Lite](#)

Installer Setup Program, [Run the NonStop CORBA OSS Installer Setup Program](#)

Installer wizard, [Host Installation Installer Wizard](#)

Internet Inter-ORB Protocol (IIOP), [IIOP and GIOP Protocol Support](#)

Interoperable Naming Service, [Naming Service](#)

IPSetup program, [Run the IPSetup Program](#)

## J

### Java Stack Example

building, [Building the Stack Sample](#)

configuring NonStop CORBA, [Configuring NonStop CORBA for the Stack Example](#)  
overview, [The Stack Example Overview](#)  
running, [Running the Java Stack Example](#)  
running on OSS, [Running the Stack Example on OSS](#)

Java Transaction Service (JTS), [Transaction Protection and Data Integrity](#), [Java Transaction Service \(JTS\)](#)

## JORB Lite

building stack example, [Building the Stack Example](#)  
calling NonStop CORBA applications from, [Calling NonStop CORBA Applications from JORB Lite](#)  
directory structure, [The JORB Lite Directory Structure](#)  
installing, [Installing JORB Lite](#)  
interoperability, [Running Clients and Servers on Remote Machines](#)  
overview, [JORB Lite Overview](#)  
package, [The JORB Lite Package](#)  
running IDL compiler, [Running the IDL Compiler](#)  
running stack example, [Running the Stack Example](#)  
stack example, [Building and Running the Stack Example](#)  
test installation, [Running JORB Lite](#)

## L

Language bindings, [Language Bindings](#)

Legacy Wrapper Interface, [Legacy Wrapper Interface](#)

## M

Migration information, [Product Level Migration Information](#)

## N

Naming Service, [Naming Service](#)

Network session concentration, [Network Session Concentration](#)

NonStop CORBA application servers, definition of, [Client/Server Computing with NonStop CORBA](#)

NonStop CORBA clients, definition of, [Client/Server Computing with NonStop CORBA](#)

NSots XID broker, [NSots XID Broker](#)

## O

Object by Value, [Object by Value](#)

Object Request Broker (ORB)

scalability, [Scalability of the Object Request Broker \(ORB\)](#)

Object Transaction Service (OTS), [Transaction Protection and Data Integrity](#)

## P

Portable Object Adapter (POA), [Portable Object Adapter](#)

## S

Scalability

application server process, [Scalability of Application Server Processes](#)

Object Request Broker, [Scalability of the Object Request Broker \(ORB\)](#)

Software and hardware requirements, [Hardware and Software Requirements](#)

Subsystem resource cleanup, [Perform Subsystem Resource Cleanup](#)

System architecture, [Why NonStop CORBA](#)

## T

Tracing, [Error Logging and Tracing](#)

Transaction protection, [Transaction Protection and Data Integrity](#)

TS/MP environment, advantages of, [Advantages of the TS/MP Environment](#)

## U

unconfigure script, [Perform Subsystem Resource Cleanup](#)

Upgrading

from NonStop CORBA 2.3 Update 3 or later to NonStop CORBA 2.6.1, [Upgrading From NonStop CORBA 2.3 to NonStop CORBA 2.6.1](#)

## W

Wrapping, [Legacy Wrapper Interface](#)